Symbiosis between quantum physics and machine learning:

Applications in data science, many-body physics and quantum computation

Vladimir Vargas Calderón

2022

SYMBIOSIS BETWEEN QUANTUM PHYSICS AND MACHINE LEARNING: APPLICATIONS IN DATA SCIENCE, MANY-BODY PHYSICS AND QUANTUM COMPUTATION

Tesis doctoral presentada como requisito para optar por el título de

Doctor en Ciencias - Física

por

Vladimir Vargas-Calderón

Agosto 2022

SIMBIOSIS ENTRE LA FÍSICA CUÁNTICA Y EL MACHINE LEARNING: APLICACIONES EN CIENCIA DE DATOS, FÍSICA DE MUCHOS CUERPOS Y COMPUTACIÓN CUÁNTICA

Tesis doctoral presentada como requisito para optar por el título de

Doctor en Ciencias - Física

por

Vladimir Vargas-Calderón

Agosto 2022

Vladimir Vargas-Calderón: *Symbiosis between quantum physics and machine learning: applications in data science, many-body physics and quantum computation* (2022)

This thesis has been developed in the:

Grupo de Superconductividad y Nanotecnología
Departamento de Física
Facultad de Ciencias
Universidad Nacional de Colombia

Main Advisor:	Ph. D. Herbert Vinck-Posada
Secondary Advisor:	Ph. D. Fabio Augusto González Osorio

ABSTRACT

This thesis explores the intersections between quantum computing, quantum physics and machine learning. In the three fields, estimating probability distributions plays a central role. In the case of quantum computing and quantum physics, a central object of study is the quantum state of a system, which encodes a probability distribution (the converse is not true, however, as a quantum state is an object that is more general than a classical probability distribution). In the case of machine learning, most of supervised and unsupervised learning tasks can be seen as estimating probability distributions from a training data set, which then can be used to predict by sampling or evaluating such probability distribution. Due to the famous curse of dimensionality, both present in machine learning but also in the natural intractability of Hilbert spaces, it has been established that quantum theory and machine learning have a lot to give and learn from each other.

The journey depicted in this thesis stems from quantum optics and its application to modelling quantum devices for quantum computation or simulation, such as quantum dots and their interaction with optical and acoustic cavities. Indeed, quantum computation has long been sought by the physics community and stands–in the collective imagination–as a "holy grail" to solve several problems in the industry and science. Of particular interest of mine is the study of quantum many-body problems themselves, which, in combination with quantum computing, establishes an interesting circular set of resources: quantum computation to study quantum systems that can be used for quantum computation.

Unfortunately, the promise of the "holy grail" of quantum computation has not materialised to date (even though there are known applications which are exponentially faster than any classical algorithm, e.g. the famous Deutsch-Jozsa algorithm), which is why the best known approaches to studying quantum physics or quantum chemistry are still classical algorithms. In particular, there are machine learning models, known as neural quantum states, that can be used to study quantum many-body problems. Neural quantum states are an application of machine learning techniques for studying quantum physics. In this thesis, we show fruitful approaches to studying ground states, steady-states and closed dynamics of quantum systems through neural quantum states.

This knowledge transfer does not only occur in one direction: quantum physics can also contribute to machine learning with quantum-inspired machine learning methods. In this thesis, we also present a framework that establishes an analogy between quantum state preparation and training, and also between quantum projective measurements and prediction. Our approach condenses classical data into the quantum state of a system. We manage to show that arbitrary probability distributions can be encoded in such a quantum state to arbitrary precision, given enough degrees of freedom of the quantum state. Moreover, we can condense arbitrarily large data sets into quantum states, which allow us to have gradient-free (actually, optimisation-free) training. This framework of ours was also put into action by implementing it on a real quantum computer for toy data sets.

Finally, I also present applications of neural quantum states and quantum-inspired generative modelling to industry problems such as the famous travelling salesman problem, for which we propose a qudit-based Hamiltonian whose ground state encodes its solution; and other problems such as the portfolio optimisation problem using tensor network generative models.

KEYWORDS

Quantum physics; machine learning; data science; quantum computing

RESUMEN

Esta tesis explora las intersecciones entre la computación cuántica, la física cuántica y el aprendizaje automático. En los tres campos, la estimación de distribuciones de probabilidad desempeña un papel central. En el caso de la computación cuántica y la física cuántica, un objeto de estudio central es el estado cuántico de un sistema, que codifica una distribución de probabilidad (sin embargo, lo contrario no es cierto, ya que un estado cuántico es un objeto más general que una distribución de probabilidad clásica). En el caso del aprendizaje automático, la mayoría de las tareas de aprendizaje supervisado y no supervisado pueden considerarse como la estimación de distribuciones de probabilidad a partir de un conjunto de datos de entrenamiento, que luego pueden utilizarse para predecir mediante el muestreo o la evaluación de dicha distribución de probabilidad. Debido a la famosa *maldición de la dimensionalidad*, presente tanto en el aprendizaje automático como en la intratabilidad natural de los espacios de Hilbert, se ha establecido que la teoría cuántica y el aprendizaje automático tienen mucho que dar y aprender la una de la otra.

El viaje descrito en esta tesis parte de la óptica cuántica y su aplicación al modelado de dispositivos cuánticos para la computación o la simulación cuánticas, como los puntos cuánticos y su interacción con cavidades ópticas y acústicas. De hecho, la comunidad de físicos lleva mucho tiempo buscando la computación cuántica y se erige–en el imaginario colectivo–como un "santo grial" para resolver varios problemas de la industria y la ciencia. De particular interés para mí es el estudio de los problemas cuánticos de muchos cuerpos, que, en combinación con la computación cuántica, establece un interesante conjunto circular de recursos: computación cuántica para estudiar sistemas cuánticos que pueden utilizarse para la computación cuántica.

Por desgracia, la promesa del "santo grial" de la computación cuántica no se ha materializado hasta la fecha (aunque se conocen aplicaciones exponencialmente más rápidas que cualquier algoritmo clásico, por ejemplo, el famoso algoritmo Deutsch-Jozsa), por lo que los enfoques más conocidos para estudiar la física o la química cuánticas siguen siendo los algoritmos clásicos. En particular, existen modelos de aprendizaje automático, conocidos como estados cuánticos neuronales, que pueden utilizarse para estudiar problemas cuánticos de muchos cuerpos. Los estados cuánticos neuronales son una aplicación de las técnicas de aprendizaje automático para estudiar la física cuántica. En esta tesis, mostramos enfoques fructíferos para estudiar estados básicos, estados estacionarios y dinámicas cerradas de sistemas cuánticos mediante estados cuánticos neuronales.

Esta transferencia de conocimientos no sólo se produce en una dirección: la física cuántica también puede contribuir al aprendizaje automático con métodos de aprendizaje automático inspirados en la cuántica. En esta tesis también presentamos un marco que establece una analogía entre la preparación del estado cuántico y el entrenamiento, y también entre las mediciones proyectivas cuánticas y la predicción. Nuestro enfoque condensa los datos clásicos en el estado cuántico de un sistema. Conseguimos demostrar que se pueden codificar distribuciones de probabilidad arbitrarias en dicho estado cuántico con una precisión arbitraria, dados suficientes grados de libertad del estado cuántico. Además, podemos condensar conjuntos de datos arbitrariamente grandes en estados cuánticos, lo que nos permite tener un entrenamiento sin gradiente (en realidad, sin optimización). Este marco nuestro también se puso en práctica implementándolo en un ordenador cuántico real para conjuntos de datos de juguete.

Por último, también presento aplicaciones de los estados cuánticos neuronales y el modelado generativo de inspiración cuántica a problemas industriales como

el famoso problema del viajante de comercio, para el que proponemos un Hamiltoniano basado en el qudit cuyo estado fundamental codifica su solución; y otros problemas como el de optimización de carteras mediante modelos generativos de redes tensoriales.

PALABRAS CLAVE

Física cuántica; machine learning; ciencia de datos; computación cuántica

AGRADECIMIENTOS

Agradezco a Herbert y Fabio por su acompañamiento, guía y amistad. A pesar de ser su estudiante, siempre he sentido de parte de ellos un trato de colega. Su camaradería y empuje contagioso, visto en uno que otro restaurante de Medellín y en cada metro de la subida a Patios, me han llenado en estos años del orgullo indescriptible de ser un hijo de la educación pública de mi país. Quiero reconocer el gran impacto que han tenido mis maestros, todos, desde los del Merani hasta los de la Universidad Nacional, en mi formación como ser humano y como investigador.

No sé si sea un accidente, o un asunto de perspectiva, pero a pesar de que la profesión de investigar está llena de frustraciones (y también de exquisitas recompensas), me ha dispuesto en un mundo lleno de gente fenomenal. Agradezco a mis compañeros y amigos con los que, durante este período doctoral, he compartido mis gustos, mis felicidades y mis tristezas, más allá de que hayan o no tenido una contribución directa con los temas de esta tesis. Por encima de sus excelentes ideas, los valoro por haberme dejado conectarme humanamente con ellos y por haberse vuelto parte de mi historia y de mi futuro; por apoyarme en mis sueños efímeros, por acompañarme en mi vaivén Browniano. Sepan que siempre intentaré apoyar sus sueños, ¡y qué mejor si coinciden con los míos!

Finalmente, agradezco a toda mi familia por siempre hacer lo posible por ponerme todas las rocas firmes y planas en frente de mi camino para que mi andar por esta vida sea ligero y parejo.

Que nazcan las hojas verdes por entre las hojas secas.

CONTENTS

1 INTRODUCTION AND MOTIVATION

"What does a physicist do?" asked Fabio, my PhD co-advisor. I thought to myself: "what a weird question." I responded "I guess physicists perform experiments to corroborate theories about fundamental stuff. These theories arise from sparks of brilliancy in which a physicist is able to abstract laws from the behaviour of nature." By the time, I had learnt the main ideas of machine learning thanks to Jorge Camargo (a former student of Fabio and co-worker of my father at a telecommunications company), and I had already started doing interesting research projects on natural language processing with him. Fabio followed his question with another: "What does machine learning do?." I immediately knew where his thoughts were leading: he wanted me to admit that, in the end, machine learning tried to do something similar to what physicists do. "Machine learning algorithms take a bunch of data and find patterns in it to solve a learning task" I said. However, I told him that I did not believe creativity could ever be mimicked from a computer. Today, I refuse to believe that, despite huge progress in generative modelling of art (music, literature with language models, paintings, etc.), computers will ever mimic true intelligence. However, it is likely that our brains are just doing a bunch of computations that could be reproducible, and that is all there is. I leave this discussion to another scenario, as I want to focus on physics and machine learning.

The aforementioned conversation took place in 2017, when I was already under the guidance of Herbert, my PhD advisor, at the Quantum Optics and Quantum Information group, and later on at the Superconductivity and Nanotechnology group of the physics department of the Universidad Nacional de Colombia. With Herbert I had started to learn about open quantum systems: real-world quantum systems are never truly isolated (though there are experimental platforms where this approximation is not bad at all), and the interaction with the environment they are embedded in must be taken into account to correctly predict properties of these quantum mechanical systems. Why exactly open quantum systems? Herbert's interests, which became my interests as well, were studying physical systems that could be used for different quantum technologies. In particular, he was very keen in studying a type of physical system (semiconductor quantum dots and their interaction with other devices) which is a promising platform to develop quantum computation, an application envisioned by the great Richard P. Feynman.

In 2019, just as I was enrolling on the PhD program that culminated in this thesis, I got to read a paper by Saito [2017], which was recommended by one of my friends at Herbert's group. For the first time, I saw a useful idea of applying machine learning for quantum physics: a neural network can be used to parameterise the wave function of a quantum system, and the parameters can be tuned so that the neural network would express the ground state of said quantum system. I started to reproduce the results from that paper, and I thought it would be very interesting to discuss this with Fabio. He told me something unbelievable: he was leading a seminar on machine learning and physics! At that time, Fabio's interests were different than mine. He wanted to use the mathematical framework of quantum mechanics to propose novel machine learning methods. One of the most interesting tools that were constantly discussed in that seminar were Penrose's tensor networks, which are widely used in physics, for instance, to find ground states of Hamiltonians through the density matrix renormalisation group method. In the seminar, a lot of interest was devoted to works that used tensor networks to capture complicated correlations in different data sets. Also, quantum computing was being intensively discussed in a more direct (a more engineering-like) manner than the quantum physical device approach that I experienced in Herbert's group.

These experiences and connections configured what have remained my academical interests since then: quantum devices for quantum computing, quantum computing itself, quantum physics for machine learning and machine learning for quantum physics. This thesis deals with these fields which lie in the intersection between quantum physics and machine learning. Figure 1.1 shows a non-accurate meme of this intersection, which, in fact, is not forceful at all. The reason why machine learning and quantum physics have a lot to give to each other is the so-called *curse of dimensionality*, a term used by the machine learning community to indicate the difficulty of learning patterns from data which live in huge vector spaces. A typical example of this is the space of an image such as the one in fig. 1.1: the image has W pixels in the horizontal direction and H in the vertical direction. Each pixel is normally represented by three colour channels: red, green and blue, each of which can have a brightness integer value from 0 to 255. This means that the image space has a size of $(3 \times 256)^{W \times H}$, where $W \times H$ is the number of pixels in the image, normally containing of the order of megapixels. It is clear that the space is intractable. Similarly, quantum physics usually deals with objects that live in intractable Hilbert spaces, whose size are also exponential with respect the number of components of the system. In both quantum physics and machine learning, science and engineering have refined methods to deal with objects described in these large spaces. Therefore, as mentioned, success of lending techniques from one area onto the other is not accidental: only benefit can be drawn from this exchange of ideas.

Now that the relation between machine learning and quantum physics is clearer, I want to focus on the exact type of problems that each one deals with (or at least the problems I'm mostly interested in). Let us start with quantum physics. A crucial problem in quantum physics is the calculation of the ground state of a physical system. Especially at low temperatures, a physical system tends to occupy its ground state (and the low-excitation states). The structure of the ground state dictates its properties, which are expected values of observables that characterise the behaviour of the physical system. Particularly in the cases of controllable quantum systems, i.e., quantum systems whose interactions can be tuned in the laboratory, it is very important to understand the ground states of such systems in different configurations of the interactions, as they will reveal behaviours that render useful for many technological applications.

To be more specific, we introduce the usual quantum mechanical formalism of a Hilbert space $\mathcal{H}$, which is the space of all possible configurations of a quantum system. Such quantum system is normally described, energetically, by a Hamiltonian $\hat{H}$, which is an operator acting on the Hilbert space, with the property that $\langle\psi|\hat{H}|\psi\rangle$ is the energy of the state $|\psi\rangle \in \mathcal{H}$. $\mathcal{H}$ is usually written as a sum of k-local

Figure 1.1: A meme which adapts to me, Herbert and Fabio from High impact PhD memes.

terms in a particular basis of the Hilbert space, which is sometimes referred to as the canonical basis. Therefore, the problem of finding the ground state of a physical system described by the Hamiltonian $\hat{H}$ is the problem of finding coefficients $\{\alpha_j\}$ such that

$$\left|\psi_g\right\rangle = \sum_j \alpha_j \left|\alpha_j\right\rangle \tag{1.1}$$

gives the global minimum of $\left\langle\psi_g\middle|\hat{H}\middle|\psi_g\right\rangle$, with a normalised state, i.e., $\sum_j \left|\alpha_j\right|^2 = 1$, where α_j is the coefficient of the ket $\left|\alpha_j\right\rangle$[1]. This means that the ground state coefficients that minimise the energy of the system induce a probability distribution over the canonical basis of the Hilbert space. This is one of the most important features of quantum mechanics and its use of complex numbers: algebra and geometry are intertwined!

As mentioned before, in reality, systems are coupled to their environment, and more complete descriptions of their physics are needed. The dynamics of these systems, under the Born-Markov approximation [Breuer et al., 2002] (my undergraduate thesis contains a complete account of this approximation [Vargas-Calderón, 2018]), is not described by the Scrödinger equation. Instead, the dynamics of the open quantum system is given by the Gorini–Kossakowski–Sudarshan–Lindblad (GKSL) equation [Lindblad, 1976; Gorini et al., 1976], which predicts the evolution of the state of the system. A vector state can no longer describe the state of the quantum system: a more general object is needed to describe classical uncertainty at the same time that we describe quantum uncertainty. The state of the quantum system is defined by a density matrix, which is given by some coefficients $\{\xi_{j,k}\}$ such that

$$\rho = \sum_{j,k} \xi_{j,k} \left|\alpha_j\right\rangle\!\left\langle\alpha_k\right| \tag{1.2}$$

is a trace-one, Hermitian, positive semi-definite operator. In contrast to the closed quantum system case, the open quantum system exchanges energy with its environment, meaning that it can gain or lose energy during its evolution. However, there is always a guaranteed steady-state [Hatano and Petrosky, 2015; Petrosky and Prigogine, 1996] that does not change over time. In the language of the GKSL equation ($\partial_t \rho = \mathcal{L}\rho$, where $\mathcal{L}$ is called the Liouvillian super-operator), one has that $\mathcal{L}\rho = 0$ for the steady-state. Similar as the ground state for the closed quantum system case, the steady-state is of paramount importance, as its structure is related to properties of the system that define its nature. Once again, the coefficients related to the steady-state also induce a probability distribution over the canonical basis of the doubled Hilbert space.

The ground-state problem and the steady-state problem become central tasks in quantum mechanics for several applications, and are, therefore, central to this thesis. Now, we will turn to a very general formulation of machine learning tasks, and will see the immediate connection with these quantum mechanical problems.

Machine learning tasks can be very different, but they all try to inferr patterns from a lot of data. In contrast with customary statistical techniques, machine learning is faced with data sets that depict complicated relations difficult to narrow down. In other words, it is difficult to assume the functional form in which variables of such data set are related. Therefore, machine learning disposes of quite general functional approximation forms that can adapt to very complicated relations between data variables.

There are two main big classes of tasks in machine learning: supervised and unsupervised learning [Bishop, 2006]. Supervised learning consists of learning how to predict a variable y given some input features $\boldsymbol{x}$ from a data set $\{(\boldsymbol{x}_i, y_i)\}$. In this set up one normally defines a class of models parameterised by some parameters $\boldsymbol{\theta}$

[1] The set of states $\{\left|\alpha_j\right\rangle\}$ is assumed to be orthonormal.

Figure 1.2: Face of a human that does not exist. This picture was generated from sampling the probability distribution of human faces encoded into a generative adversarial network [Karras et al., 2019]. Taken from thispersondoesnotexist.com.

such that the class of models $f_{\boldsymbol{\theta}}$ is able to predict with great accuracy the dependent variable, i.e., $f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) \approx y_i$. The process of finding an optimal set of parameters $\boldsymbol{\theta}$ is called training, and usually consists of minimising a cost function. For instance, for regression problems one can use a cost function such as $\sum_i |f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i|^2$. On the other hand, unsupervised learning is concerned with 'untagged' data, meaning that the dataset consists only of input features $\{\boldsymbol{x}_i\}$. These features are assumed to be drawn from a probability distribution $P(\boldsymbol{x})$ that is to be approximated through a machine learning model. Therefore, the approximated probability distribution is also parameterised by some parameters $\boldsymbol{\theta}$, and looks like $P_{\boldsymbol{\theta}}(\boldsymbol{x})$ (these are probability distributions over spaces that can be intractable, such as the space of images and the probability distribution of human faces, as shown in fig. 1.2). There are different flavours to this general view of unsupervised learning, but, overall, they all consist of the same: estimating a probability distribution. In fact, one can see supervised learning as a probability distribution estimation problem as well. One can consider learning the joint probability distribution $P(\boldsymbol{x}, y)$ and perform predictions as $P(y|\boldsymbol{x})$.

The latest remarks on supervised and unsupervised learning build a clear bridge with the quantum physics problems of finding ground-states or steady-states. This means that a machine learning model can parameterise wave functions or density matrices. More explicitly, the coefficients α_j in eq. (1.1), or the coefficients $\zeta_{j,k}$ in eq. (1.2), can be predicted by a machine learning model. A common choice for doing this are artificial neural networks. These are layered models that combine linear operations with non-linear transformations[2].

Even though the young community in the intersection between quantum physics and machine learning has focused on using neural networks, we emphasise that there are many other machine learning models that can be used to estimate proba-

[2] The most widely used neural network is the full-forward neural network, which defines a hidden state as

$$\boldsymbol{h}_i = f_i(W_i \boldsymbol{h}_{i-1} + \boldsymbol{b}_i). \tag{1.3}$$

Here, $\boldsymbol{h}_i$ is the hidden state of the i-th layer, which results from performing a linear transformation of the $i-1$-th hidden state through a matrix W_i, adding a bias term $\boldsymbol{b}_i$ and computing an element-wise non-linear function f_i. The first hidden state is not hidden per se, and corresponds to the input features $\boldsymbol{h}_0 = \boldsymbol{x}$. The last hidden state corresponds to the output y.

bility distributions. Moreover, as I will show in this thesis, from the perspective of quantum physics one can also propose machine learning models that take advantage of the mathematical framework of quantum mechanics to express probability density functions.

Before exposing the main quests of this thesis, I want to address a very important topic which also lies at the heart of the intersection between quantum physics and machine learning: quantum machine learning. Quantum computation is the field of computing with quantum information, i.e., computing using the state of quantum systems [Nielsen and Chuang, 2002]. Long story short, quantum computing is a tool that has the potential to unveil unprecedented computing power to solve central problems of humanity. However, the experimental challenge of building a useful "fault-tolerant" quantum computer is huge, and is taking the effort of thousands of physicists and engineers around the world, both in academy but also in the ever-growing quantum computing industry. So far, the quantum computers that have been built are not fault-tolerant, meaning that levels of noise and decoherence significantly damage the quantum information processed in quantum computers. These devices are known as near-term intermediate-scale quantum computers (NISQ) [Bharti et al., 2022].

A particular set of quantum algorithms, variational quantum algorithms, is regarded as the most promising area to achieve practical quantum advantage [Cerezo et al., 2021] in the NISQ era, which differs from the famous quantum supremacy experiments that have already demonstrated that quantum computers, even in the NISQ era, are able to perform some computations that are not realisable with classical computers [Arute et al., 2019; Zhong et al., 2020; Madsen et al., 2022], but are far from useful for science and technology[3]. Practical quantum advantage involves problems that are actually useful in terms of further developing science (in Feynman's view, quantum computers can help the study of quantum mechanics itself [Feynman, 1982]), or developing technology. Such problems are the ones tackled by regular machine learning, but also by quantum machine learning, which is synonym to these variational quantum algorithms. Indeed, variational quantum algorithms are based on variational quantum circuits, which are also known as quantum neural networks. Variational quantum circuits are quantum circuits with a given set of (generally), one- and two-qubit gates that induce general rotations whose angles are treated as parameters that can be tuned to minimise a cost function that is calculated using the state of the quantum circuit after the quantum gates have been applied. A great deal of research is currently being put forward on variational quantum circuits, but achieving practical quantum advantage is currently hampered by several reasons. First, the study of variational quantum circuits has to be done, today, using classical computers, which limits the size of the circuits that can be simulated. In these studies, exact access to the state of the quantum circuit is assumed, but in a real quantum device this is not possible. Second, in order to compute any quantity using the state of the quantum circuit, one needs to measure the circuit a large number of times to *estimate* said quantity. Finally, learning, as in regular neural networks, occurs through gradient-based algorithms. The estimation of these gradients usually require further measurements of the quantum circuit, which impose an important overhead. All these disadvantages are also accompanied by a much more fundamental issue: training variational quantum circuits portrays different theoretical problems [Anschuetz and Kiani, 2022] such as the onset of barren plateaus [McClean et al., 2018]. Nonetheless, the scientific community is investing great resources to sort these issues out, or to find ways to bypass them.

For all these reasons, I believe that classical computing will continue (for a while) to provide the best possible numerical methods to study optimisation problems and quantum mechanical problems. Nonetheless, the development of quantum physics (with its plethora of applications), machine learning and quantum computation

[3] Actually, Google's supremacy experiment has been already dismantled by classical computation [Pan et al., 2021]

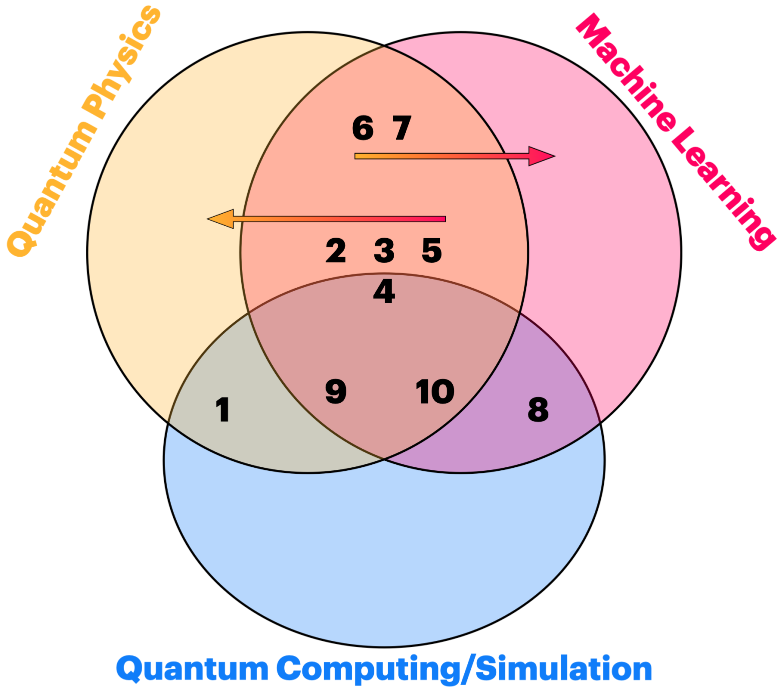

Figure 1.3: Depiction of the different works carried out during my PhD work, all of which lie in the intersections between quantum physics, machine learning and quantum computing and simulation. The numbers refer to different works which are explained in detail in the main text. The arrows indicate that there are two groups of works which particularly exploit quantum physics for machine learning (works 6 and 7), and machine learning for quantum physics (works 2, 3, 4 and 5).

is going and will go hand in hand in the near future: progress in any of these fields will have implications in the others as the niche of people that work in this intersection continues to grow. All of these topics, of dear interest to me, interleave in this thesis. It must be understood that this thesis, unlike the usual doctoral thesis, does not dwell on a singular problem, but is rather the result of the *exploration*[4] of these fields and their interactions. This is depicted in fig. 1.3, which now I explain in detail.

First, it is important to understand the common object that is treated throughout the thesis: (quantum) probability distributions. "Quantum" is in parenthesis because most of the time we will deal with quantum states (pure or mixed), which induce a probability distribution. We (you, the reader and I) will be concerned with estimating, learning or calculating a quantum state of any sort at every single section of this thesis, which will serve to study quantum physics, quantum computation or applications to data science.

This thesis is structured in chapters that develop the ten works shown in fig. 1.3. Each chapter will present its own brief state of the art. An introduction to these chapters, along with an introduction to the works is given next.

Chapter 2— The physical implementation of quantum computers/simulators requires understanding and proposing quantum control techniques for different experimental physical platforms. Here, we investigate–in collaboration with José Maria Villas-Bôas– an experimental platform that is a strong candidate for quantum computation: semiconductor quantum dots. They are normally used as qubits because single excitations (electron-hole pairs known as excitons) can be isolated. Herbert's group has a great tradition of studying light-matter interactions in these kinds of systems, as light-matter interactions are a vehicle to control the state of

[4] Exploration in the sense that we do not particularly focus on exploiting a technique, but we explore how one field can lend ideas to the other fields.

qubits by sending pulses of light. Lately, we have also been taking into consideration matter-matter interactions through phonons in the dissipative case (my undergraduate work revolved around this [Vargas-Calderón and Vinck-Posada, 2019]), as it is necessary to understand how lattice vibrations can damage quantum information stored in semiconductor quantum dot-based qubits [Vargas-Calderón and Vinck-Posada, 2020]. Nonetheless, the development of acoustic cavities leverage the possibility of coherently coupling phonons with excitons.

We studied an optomechanical setup where we were able to theoretically demonstrate that dark excitons could be coupled with N-phonon bundle states in an acoustic cavity, which, due to a dissipative channel, was able to emit those N-phonon bundles. This work (**Work 1** in fig. 1.3) contributes to the understanding of multi-phonon physics and puts on the table a tool that can be used in quantum information protocols for quantum computation. This work is presented in the paper "Dark Exciton Giant Rabi Oscillations with no External Magnetic Field" [Vargas-Calderón et al., 2022].

Chapter 3— This chapter explores the right-to-left arrow shown in fig. 1.3, which includes **Works 2, 3, 4 & 5**. We tackle a similar problem to the one considered in chapter 2: we try finding ground-states and steady-states for lattices of interacting systems. However, due to the size of the systems, we can no longer use exact diagonalisation methods. Usually, large lattices of interacting systems are theoretically treated using mean-field approximations. These approximations fall short to explain different phenomena. We use neural quantum states to go well-beyond such an approach using the so-called variational Monte Carlo method [Becca and Sorella, 2017] that allows us to handle intractable Hilbert spaces such as the ones characteristic of large bosonic lattices. A big portion of my PhD was devoted to the development of numerical methods for finding ground states and steady-states. This was particularly done through my participation of the NetKet project as a collaborator, which is an international effort to build and maintain a Python library for large-scale and high-performance computing of neural quantum states [Carleo et al., 2019a]. Our collaboration has shown this work (**Work 2** of fig. 1.3) in the paper "Netket 3: Machine learning toolbox for many-body quantum systems" [Vicentini et al., 2021]. NetKet has become the *de facto* tool for research in neural quantum states and the main tool for research in this chapter of the thesis.

With such a tool (and the familiarity only given by getting to know the inside of the library), we turned our attention to one of the most interesting phenomena in physics: phase transitions, which witness macroscopic changes in the behaviour of physical systems. In quantum mechanics, phase transitions are characterised by structural changes in the ground state of a closed system, or structural changes in the steady-state of an open system [Minganti et al., 2018]. As an example, we take as benchmark the widely known Bose-Hubbard model, which describes interacting bosons that can occupy sites in a lattice. We study the phase transition of this system, which involves the Mott insulator and superfluid phases, through the novel neural quantum state approach whereby the wave function or density matrix are described by neural networks.

As mentioned, with Netket, we have studied the Bose-Hubbard model from different perspectives. We reconstructed the phase-diagram of the Bose-Hubbard model for small lattices looking for differences between neural quantum states and exact diagonalisation states, finding that, in general, neural quantum states provide great accuracy for measuring observables of the Bose-Hubbard system, which show the Mott-insulator-to-superfluid phase transition. However, near the phase boundary, differences between exact states and neural quantum states become significant. We reported these results in our work (**Work 3** in fig. 1.3) "Phase diagram reconstruction of the Bose-Hubbard model with a Restricted Boltzmann Machine Wavefunction" [Vargas-Calderón et al., 2020], which was also presented at the Quantum Techniques in Machine Learning conference in 2020[5].

[5] https://www.zapatacomputing.com/events/qtml-2020/

In collaboration with Juan Pablo Restrepo Cuartas, another PhD student at Herbert's group, by treating the Bose-Hubbard Hamiltonian in the momentum space, we show that neural quantum states are also able to find the ground state in the superfluid phase. This work was presented at the 2nd Colombian Meeting on Many-Body Quantum Simulation under the title "Machine Learning the ground state of the Bose-Hubbard model in momentum representation" in 2021[6].

We also used neural quantum states to study the time evolution of a system through the Feynman-Kitaev formalism, which converts a time evolution problem into a ground state problem in the work (**Work 4** in fig. 1.3) "An Empirical Study of Quantum Dynamics as a Ground State Problem with Neural Quantum States" [Vargas-Calderón et al., 2022]. By studying the exact ground state structure, we use the Feynman-Kitaev formalism to create ever more difficult problems for the neural quantum states. We conclude that neural quantum states are limited by trainability, even though they are able to express the desired ground state. In other words, neural quantum states are able to correctly describe the ground state of the Feynman-Kitaev system, but learning the neural network parameters becomes difficult as the true ground state becomes entangled and spread through the Hilbert space.

The difficulties of training neural quantum states are also seen on extensive experiments on the driven-dissipative Bose-Hubbard system, which considers a coherent boson pump as well as a dissipative boson escape mechanism (**Work 5** of fig. 1.3). The dissipation mechanism is further explored in two different setups. In the first setup dissipation occurs at every site of the Bose-Hubbard lattice. The second setup refers to localised dissipation, in which one can engineer dissipation at a particular site of the lattice, causing controllable escape of particles, or controllable introduction of dephasing. We find a considerable difficulty in learning the steady-state of this system, but we envision a path forward.

Chapter 4— The highway between quantum physics and machine learning is now explored in the other direction. In chapter 3 we saw that neural quantum states are neural networks from machine learning leveraged to solve quantum physics problems. Now we ask ourselves how can quantum physics, or rather, the mathematical framework of quantum physics help solving problems in machine learning. This is summarised in the left-to-right arrow that goes from quantum physics to machine learning in fig. 1.3, which contains **Works 6 & 7**.

These works started to take form from my participation at the physics-machine learning seminar led by Fabio, which decanted in discussions that established an analogy between the experimental realisation of experiments in quantum physics and the usual steps that one follows when doing machine learning. Let me explain. In machine learning, one trains an algorithm using a data set, and then uses the trained machine learning model to make predictions. Fabio compared the process of training to the process of preparing a quantum state that encoded classical data and the correlations in that data. Then, the process of prediction was equated to performing a projective measurement using the previous quantum state. The previous analogies are not merely for depiction. After plenty of discussion, we took advantage of the concept of superposition in quantum mechanics to be able to condense an arbitrarily large data set into the quantum state of a system (e.g. $|\Psi\rangle = \sum_i |\psi(x_i)\rangle \otimes |\phi(y_i)\rangle$, where $|\psi(x_i)\rangle \otimes |\psi(y_i)\rangle$ are states corresponding to individual data points (x_i, y_i)). From such a data set quantum state, one can perform projective measurements onto the x part of the quantum system, leaving a reduced density matrix on the y part, which will contain the probabilities that each set of features x assumes a value y.

This idea was used first to perform classification of data in our work (**Work 6** in fig. 1.3)"Classification with quantum measurements" [González et al., 2021b]. Later on, in collaboration with other PhD students from Fabio's group, the idea was generalised to other learning tasks such as kernel density estimation, kernel

[6] https://sites.google.com/correounivalle.edu.co/2ndcol-meeting-many-body-simul/home

density classification, and quantum measurement regression. This work is pending publication in the Quantum Machine Intelligence journal under the title "Learning with Density Matrices" [González et al., 2021a] (**Work 7** in fig. 1.3), and was also presented at the Quantum Techniques in Machine Learning conference in 2021[7].

Since the mentioned machine learning models are based on ideas from quantum mechanics, it was only natural to adapt them to actual quantum computers. For this reason, a Python library was written to create quantum circuits where data set states were prepared, and projective measurements were made in order to perform density estimation and classification on real quantum devices from IBM. Results from this work are presented in (**Work 8** in fig. 1.3) "Optimisation-free Density Estimation and Classification with Quantum Circuits" [Vargas-Calderón et al., 2022], which were presented in the works "Optimisation-free classification with quantum circuit measurements" and "Non-parametric multivariate continuous density estimation in a quantum computer" at the Quantum Techniques in Machine Learning conference in 2021[7].

Chapter 5— The last chapter of this thesis is focused on applications of the aforementioned methods to the industry[8]. By combining the neural quantum state approach, along with Nicolás Parra, another PhD student at Herbert's group and a dear friend of mine, we show that we can find ground states that encode solutions of combinatorial problems that have been historically studied for their applications to industry, such as the travelling salesman problem. In "Many-Qudit Representation for the Travelling Salesman Problem Optimisation" [Vargas-Calderón et al., 2021] (**Work 9** in fig. 1.3), which was also presented at the Quantum Technology International Conference[9], we mapped the famous travelling salesman problem to the Hamiltonian of a many-qudit system, and find its solution through ground state search. We compare the solvability of the travelling salesman problem using our qudit map to the usual approach which involves mapping the problem to a quadratic unconstrained binary optimisation problem, that can be solved with a qubit simulator. It was found that our approach was easier to solve using neural quantum states, as expected, which motivates the development of quantum simulation based on qudits. This work involved the largest simulation that we have carried out, where the energy of a system composed of 1600 qubits was minimisied.

Finally, we also give a short discussion of Zapata Computing's Generator-Enhanced Optimisation © [Alcazar et al., 2021], which is a proprietary framework for which I developed software for solving optimisation problems based on the generation of solutions through classical and quantum generative models (**Work 10**[10] in fig. 1.3). This was part of my PhD internship at this company, which specialises in bringing quantum-ready applications™ to its customers.

To conclude, this thesis will explore different ideas found in the conversation between quantum physics, quantum computing/simulation and machine learning. In chapter 6 I provide some conclusions and list in more succinct form the contributions of this thesis.

[7] https://www.quantummachinelearning.org/qtml2021.html

[8] I am now part of the evolving quantum computing/information/machine learning industry thanks to my involvement in Zapata Computing Inc.

[9] https://premc.org/conferences/qtech-quantum-technology/

[10] I am not an author of this work, though I have been the main developer of a project that leverages the Generator-Enhanced Optimisation for large-scale high-performance computing. Other efforts which include a continuous version of it have also been developed under my lead, but I do not present those results in this thesis due to patent filing paper-work that is due. Despite this, the general ideas of Generator-Enhanced Optimisation prevail and are fully shown in this thesis.

2 PHYSICS OF QUANTUM DEVICES FOR QUANTUM INFORMATION PROCESSING

In this chapter we will focus on multi-phonon physics, which deals with the manipulation and control of quantum physical processes which involve many phonons. Why? We will see in the chapter that this will constitute an option for controlling the state of many-level systems or qudits for quantum computing or quantum simulation. We will investigate a particular experimental set up which involves quantum dots embedded in layers that interact with surface acoustic waves, and which can be pumped coherently by fine-tuned lasers (see fig. 2.1). This set up is particularly interesting because experiments are almost there, so our results become more relevant.

2.1 STATE OF THE ART

Let us first recall what a Rabi oscillation is. The Rabi model in quantum optics [Gerry et al., 2005; Fox et al., 2006] is a rather simple semi-classical model of the interaction between a two-level system (e.g. a bright exciton in a quantum dot) and a classical field of light. The two levels of the system are the ground state $|g\rangle$ and an exciton state $|e\rangle$. The energy of the two-level system is,

$$H_A = \hbar\omega_X \, |e\rangle\langle e| \, . \tag{2.1}$$

The interaction with the classical field of light is given by an dipole interaction term of the form $-\boldsymbol{d}\cdot\boldsymbol{E}$, where the electric field $\boldsymbol{E}$ is given by a forward and a backward waves of the form $\Omega e^{\pm i\omega_c t}$. By considering the rotating wave approximation [Gerry et al., 2005], one can write the dipole interaction term as

$$H_{AF} = \frac{\hbar\Omega}{2}(\sigma e^{i\omega_c t} + \sigma^\dagger e^{-i\omega_c t}), \tag{2.2}$$

where $\sigma := |g\rangle\langle e|$ is the exciton lowering operator. The system described by the Hamiltonian $H_A + H_{AF}$ will undergo oscillations between the $|g\rangle$ state and the $|e\rangle$

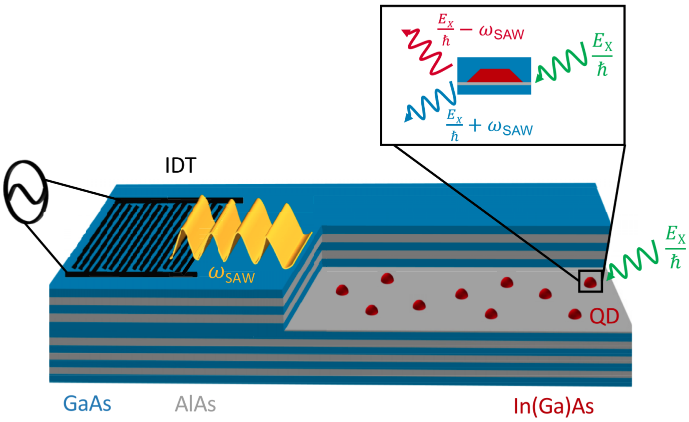

Figure 2.1: Schematic drawing of semiconductor quantum dots embedded in a surface acoustic wave resonator device, and pumped by an external laser. Taken from [Wigger et al., 2021].

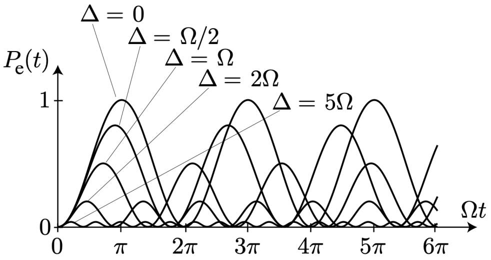

Figure 2.2: Probability of the system being excited when initialised at the ground state, depending on the detuning value $\Delta = \omega_c - \omega_X$. Taken from [Steck, 2007].

state, known as Rabi oscillations. Most interestingly, the oscillation amplitudes will depend on the value of detuning $\Delta = \omega_c - \omega_X$ between the frequency of the laser w_c and the exciton frequency ω_X, as shown in fig. 2.2. Precisely at resonance, the excited and ground state will have the same energy, and the population transfer will be maximised. This sole point is the whole key to producing Rabi oscillations in any system. Now, let us turn to the actual experimental platform that interests us.

Semiconductor quantum dots are one of the candidate platforms for quantum computing due to the possibility of treating them as effective two-level systems such as the ones we discussed previously. As shown in fig. 2.3, they are usually formed as material impurities in the bulk of a semiconductor material, which confines charge carriers within a very small volume that can be seen as a three dimensional quantum well. The excitation that is considered in a quantum dot to form the two-level system is the so-called exciton, formed by the Coulomb interaction between an electron and a hole resulting from a jump by the electron from the valence band to the conduction band. Both the electron and hole have different effective masses, and their movement through the lens-like quantum dot is spatially restricted in a different manner [Barker and O'Reilly, 2000]. The electron and the hole are confined in separate parts of the quantum dot, which makes the exciton form an electric dipole.

However, the so-called exchange interaction in a quantum dot produces a fine-structure of the excitonic energies: electron-hole pairs with anti-parallel spins have different energy from electron-hole pairs with parallel spins [Nomura et al., 1994; Bayer et al., 2002]. Moreover, due to selection rules, electron-hole pairs with parallel spins cannot trivially recombine to emit a photon. Thus, they receive the name of

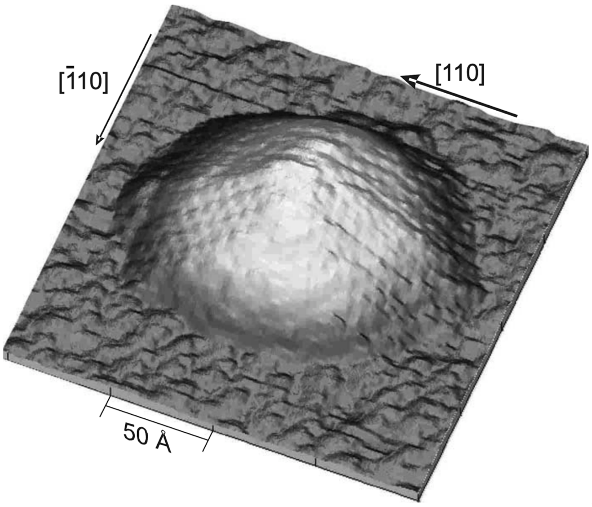

Figure 2.3: Scanning tunnelling microscopy of an InAs/GaAs quantum dot. Taken from [Eisele et al., 2008]

dark excitons. The literature normally sweeps dark excitons under the rug: a "normal" bright exciton can be formed, and then, because of relaxation processes and interaction with particles such as phonons, random spin flips may occur, converting the bright exciton into a dark exciton. The opposite process can occur, converting the dark exciton into the bright exciton. Flipping between bright and dark excitons effectively increases the exciton lifetime, but the dynamics are certainly a bit more complicated than this effective behaviour which allows people to treat an excitonic system as a two-level system, instead of the "truer" five-level system that it is: a ground state with no exciton, two bright exciton and two dark exciton states.

This quantum dot structure was ignored in Ref. [Bin et al., 2020], which is a seminal paper for multi-phonon physics in a setup similar to the one that we present in this chapter. Bin et al. [2020] consider a two-level system interacting with an acoustic cavity, and pumped by an external laser, which is similar to our setup, with the difference that we consider a five-level system. They showed how by fine tuning the laser's frequency, one is able to perform giant Rabi oscillations between a state of zero phonons and a state of N phonons. Further, considering dissipative channels such as exciton recombination due to spontaneous emission and phonon escape from the cavity, they showed that the system is able to emit N-phonon bundles.

We will see that in our case we are also able to perform giant Rabi oscillations and emission of N-phonon bundles, but we particularly couple to dark excitonic states. The reason behind this is that dark excitons, as mentioned before, are more robust because they do not recombine due to spontaneous emission, but they require a random spin flip before. However, we face the technical problem of using the laser to couple to dark excitons, which at first seems counter-intuitive because dark excitons do not interact with light fields. Therefore, we need to transfer the population from bright excitons to dark excitons.

The control of population transfer to bright and dark excitons is usually achieved by means of an external magnetic field [Neumann et al., 2021]. Herbert's group has significant experience in studying this problem, especially with the doctoral work of Carlos Jiménez and Milton Linares [Jiménez-Orjuela et al., 2017; Jiménez-Orjuela et al., 2018, 2020b,a; Linares et al., 2021]. An external magnetic field creates a

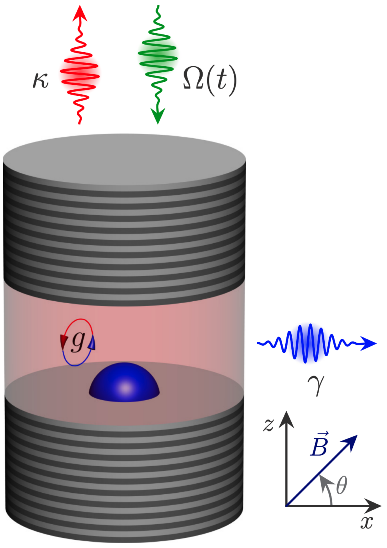

Figure 2.4: Quantum dot embedded in a micropillar acting as a double Bragg reflector that forms an optical cavity with an external magnetic field. Mechanisms of photon escape with a rate κ, coherent photon pumping from an external laser $\Omega(t)$, exciton recombination rate γ and light-matter coupling g are shown. Taken from [Jiménez-Orjuela et al., 2017].

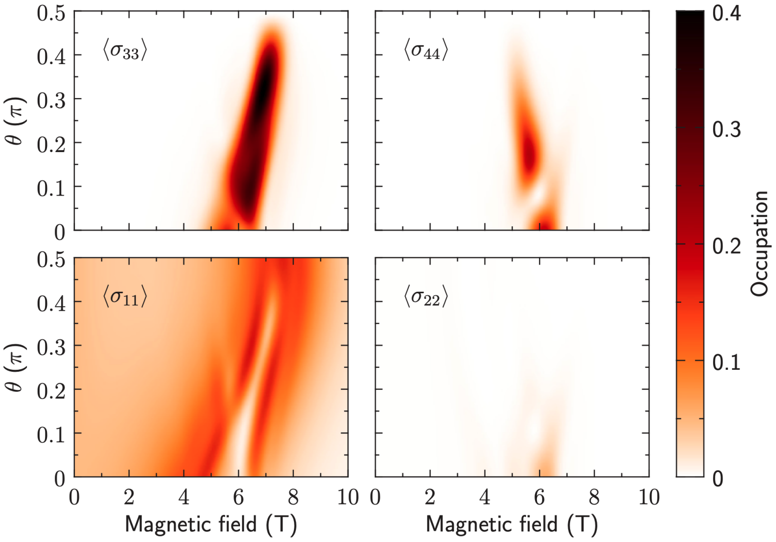

Figure 2.5: Occupation of dark exciton states as a function of the external magnetic field tilt and strength. $\langle\sigma_{11}\rangle$, $\langle\sigma_{22}\rangle$, $\langle\sigma_{33}\rangle$, and $\langle\sigma_{44}\rangle$ are the two bright excitons and the two dark excitons populations, respectively. The plots are occupations taken a short time after a laser pulse has struck the quantum dot. Taken from [Jiménez-Orjuela et al., 2017], where more details can be found.

coherent interaction between bright and dark exciton states, allowing for population transfer channels between them. A depiction of such a system can be seen in fig. 2.4.

The magnetic field can be tuned to maximise population transfer to dark excitons which suffer a much smaller effective exciton recombination rate than bright excitons. This effect can be seen in the occupation of each exciton state, shown in fig. 2.5. The quantum dot is illuminated with a laser of a certain polarity that pumps energy into the first bright exciton state for a finite amount of time. After some time passes, and evolution of the exciton-optical microcavity takes place, occupations of excitonic states are measured, signalling that an important population transfer towards one of the dark exciton states was carried out.

We will see that we do not need an external magnetic field to achieve population transfer to dark exciton states next.

2.2 RECIPE FOR GIANT RABI OSCILLATIONS

This section will give a general recipe for creating giant Rabi oscillations. From now on, in this chapter, results and excerpts of Ref. [Vargas-Calderón et al., 2022] will be shown.

With the advent of phonon control physics and the realisation of acoustic cavities, we noticed that dark exciton population could also be controlled using population transfer channels allowed by phonon interactions. In other words, in the work shown in this chapter we avoid using of a magnetic field, which can be expensive and experimentally challenging [Adambukulam et al., 2021]. In particular, we consider the Bir-Pikus interaction [Bir et al., 1974], which takes into account the interactions induced between exciton states due to a coherent intervention of phonons of the strained lattice [Pikus and Bir, 1971]: strain/stress comes from the material mismatch (the lattice mismatch) at the material boundary between the quantum dot and its surroundings. However, not only do we aim to control the population of dark exciton states, but we also consider the problem of phonon emission from an open quantum system perspective.

Particularly, we consider the physical system portrayed in fig. 2.1. A quantum dot is immersed in an acoustic single-mode cavity such as in the experiment by Weiß and Krenner [2018]. Also, an external laser is used to coherently pump the quantum dot with a constant tone whose frequency can be tuned piezoelectrically [Wigger et al., 2021]. The phonon mode will couple to the exciton states through the Bir-Pikus interaction, and the dissipative channels will describe the way in which particles escape the driven-dissipative physical system at hand. As we will see, we were able to generate giant Rabi oscillations between the vacuum of the system and a dark exciton state with N-phonons. Harvesting these giant Rabi oscillations were also useful for creating the emission of N-phonon bundles, which are important resources from multi-phonon physics.

Let us tackle the general problem of creating giant Rabi oscillations. For this, consider a system that is not composed only of two-levels, but is composed of many more (even an infinite amount of levels), can also be tuned through an external laser frequency in order to match the energy of two bare states, bringing them to resonance. By initialising the system in one of the matched bare states, the evolution of its dynamics will be restrained to the resonant states. Let me explain how this works in the case that interests us the most, i.e., the quantum dot embedded in an acoustic cavity and pumped by an external laser, as shown in fig. 2.1.

The Hamiltonian that we consider is of the following form:

$$H = H_{\text{QD}} + H_{\text{laser}} + H_{\text{cav}} + H_{\text{el-ph}}. \tag{2.3}$$

The quantum dot Hamiltonian is ($\hbar = 1$):

$$H_{\text{QD}} = \omega_X(\sigma_{11} + \sigma_{22}) + \omega_d(\sigma_{33} + \sigma_{44}) + \frac{\delta_1}{2}(\sigma_{12} + \sigma_{21}) + \frac{\delta_2}{2}(\sigma_{34} + \sigma_{43}), \tag{2.4}$$

where the bright excitons have energy ω_X, and the dark excitons have a shifted energy $\omega_d = \omega_X - \delta_0$. Moreover, exchange interaction gives rise to coupling between the bright states and coupling between the dark states through rates δ_1 and δ_2, respectively. The operators σ_{ij} are defined as the ladder operators $|i\rangle\langle j|$ for possible states being: a valence state $|v\rangle$, two bright exciton states $|1\rangle$ and $|2\rangle$ with anti-parallel electron-hole spins, and two dark exciton states $|3\rangle$ and $|4\rangle$ with parallel electron-hole spins. The laser Hamiltonian is given by

$$H_{\text{laser}} = \Omega_1(e^{-i\omega_L t}\sigma_{1v} + e^{i\omega_L t}\sigma_{v1}) + \Omega_2(e^{-i\omega_L t}\sigma_{2v} + e^{i\omega_L t}\sigma_{v2}), \tag{2.5}$$

where the laser can be polarised to pump both bright excitons with different amplitudes Ω_1 and Ω_2 at a laser frequency ω_L that can be fine-tuned piezoelectrically. The time dependence of eq. (2.5) can be removed by performing a unitary transformation $H \to UHU^\dagger + i(\partial_t U)U^\dagger$ [Steck, 2007] with $U = \exp(i\omega_L t[\sigma_{11} + \sigma_{22} + \sigma_{33} + \sigma_{44}])$. The acoustic cavity Hamiltonian is simplified to a single mode because we consider a model where this is the only mode that interacts with the excitons in the quantum dot. Hence, this Hamiltonian is given by

$$H_{\text{cav}} = \omega_b b^\dagger b \tag{2.6}$$

where b is the annihilation bosonic operator $b = \sum_n \sqrt{n}\,|n-1\rangle\langle n|$, and ω_b is the energy of the confined phonon mode. Finally, we introduce the electron-phonon interaction given by the Bir-Pikus interaction [Bir et al., 1974; Woods et al., 2004]:

$$\begin{aligned} H_{\text{el-ph}} = \Big\{ &\frac{g_{\text{bd}}}{\sqrt{2}}\left[(1+i)(\sigma_{13} + \sigma_{14}) + (1-i)(\sigma_{23} + \sigma_{24})\right] \\ &+ g_{\text{bb}}[\sigma_{11} + \sigma_{22} + i(\sigma_{12} - \sigma_{21})]\Big\}(b^\dagger + b) + \text{H.c.} \end{aligned} \tag{2.7}$$

where $g_{\text{bb(bd)}}$ are bright-bright(bright-dark) exciton coupling rates through phonons, which results from the coupling of the hole spin to the strain tensor of the quantum dot as well as the electron-hole exchange interaction [Takagahara, 1993].

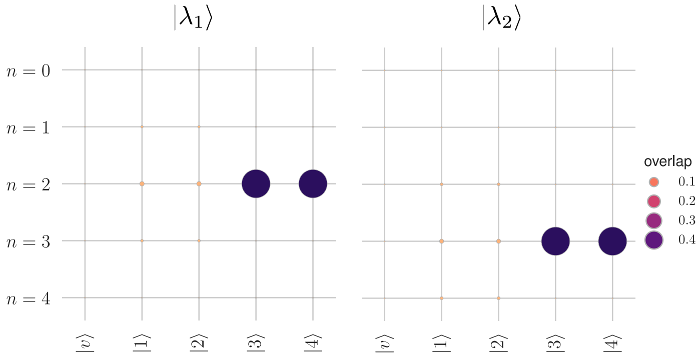

Figure 2.6: Eigenstates of eq. (2.3). The state $|\lambda_{1(2)}\rangle$ is mostly composed by the state $|2(3), d_+\rangle$, where $|d_\pm\rangle = (|3\rangle \pm |4\rangle)/\sqrt{2}$ is the dark exciton symmetric $(+)$ or anti-symmetric $(-)$ state. Taken from [Vargas-Calderón et al., 2022].

The spectrum of the complete Hamiltonian in eq. (2.3) is rather complicated in terms of all the Hamiltonian parameters. However, it is worth highlighting that the laser energy can be fine-tuned in order to leave some states in resonance. All of the interactions of eq. (2.3) are generally small in comparison to the higher energy terms in H_{QD} and H_{cav}, which count the number of excitations on the cavity and the quantum dot and adds them up with the respective energy of each excitation. This is important because in this limit of very small interaction energies, the Hamiltonian can be seen as

$$\omega_X(\sigma_{11} + \sigma_{22}) + \omega_d(\sigma_{33} + \sigma_{44}) + \omega_b b^\dagger b \tag{2.8}$$

with eigenstates $|n, \text{QD}\rangle$, with n phonons and some state for the quantum dot. The laser will essentially shift (in the laser rotating frame) the quantum dot's energies, so that $|0, v\rangle$ can be put in resonance with another state $|n, d\rangle$, where $|d\rangle$ is some combination of dark exciton states. When the states are put in resonance, they are no longer eigenstates of the Hamiltonian by themselves. Instead, a linear combination of them will be eigenstates. Numerically, the eigenspectrum were calculated constructing the Hamiltonian shown in eq. (2.3) using QuTiP [Johansson et al., 2013]. When the small interactions between phonons and excitons are turned on, the eigenstates are no longer the basis states $|n, \text{QD}\rangle$, but perturbations of them. For example, fig. 2.6 shows the tomography of a couple of eigenstates of eq. (2.3). The eigenstate $|\lambda_1\rangle$ is approximately $|2, d_+\rangle$, i.e., the dark exciton symmetric state with two phonons. Notice that it is not exactly this state, as there is a contribution from bright states and two phonons, as well as a smaller contribution from bright states and one and three phonons. Something similar happens with the eigenstate $|\lambda_2\rangle$, which largely approximates $|3, d_+\rangle$.

All that is needed to produce Rabi oscillations between two different states, or giant Rabi oscillations, which are just Rabi oscillations between two states that differ in more than one excitation, is to tune the laser so that these two states are in resonance, forming a linear combination that is an eigenstate of the whole Hamiltonian eq. (2.3). If we do this, we can produce giant Rabi oscillations on demand, as we showed in fig. 2.7. Details for the parameter values in order to reproduce the results are found in Ref. [Vargas-Calderón et al., 2022].

In fig. 2.7 one can see that the oscillations are almost perfectly achieved between the vacuum state, which has zero phonons and the quantum dot is in the valence state, and a higher-excited state with a lot of population in the dark exciton states. The frequency of oscillation, however, becomes larger and larger as the number of phonons in the targeted state increase.

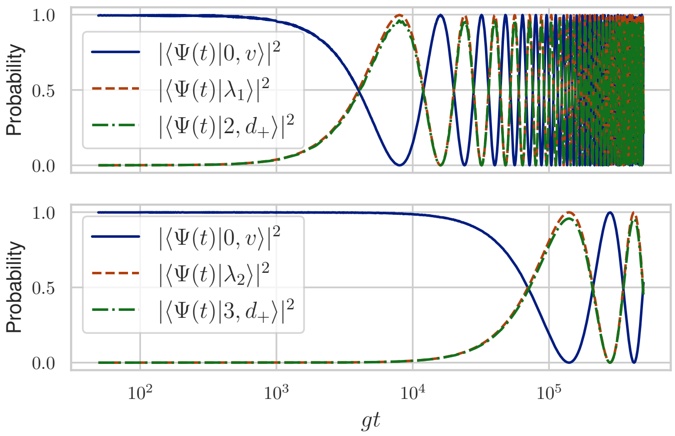

Figure 2.7: Giant Rabi oscillations between the 0-phonon valence state (solid line) and the states shown in fig. 2.6 (dashed line). The dash-dotted line shows the evolution of the $|n, d_+\rangle$ state for $n = 2$ (top panel) and $n = 3$ (bottom panel). Taken from [Vargas-Calderón et al., 2022].

2.3 QUANTUM TRAJECTORIES AND THE EMISSION OF N-PHONON BUNDLES

Real quantum systems are open quantum systems. One of the most accurate open quantum systems theories is the so-called master equation [Breuer et al., 2002], or the GKSL equation, which was introduced in chapter 1 which comes from considering the main system of interest immersed in a larger system referred to as a bath, or as the environment. The main system of interest is assumed to be separable from the environment at the beginning of time (Born approximation), and the information flowing from the main system towards the environment is not allowed to backflow into the system (Markov approximation), which result in rapidly decaying correlation functions of the bath. Details of the derivation of the GKSL equation can be found on the excellent book by Breuer et al. [2002], or in my undergraduate thesis [Vargas-Calderón, 2018].

The GKSL equation is ($\hbar = 1$)

$$\frac{d\rho}{dt} = i[\rho, H] - \frac{1}{2}\sum_m \gamma_m [A_m^\dagger A_m \rho + \rho A_m^\dagger A_m - 2A_m \rho A_m^\dagger], \tag{2.9}$$

where the operators A_m are known as collapse or jump operators, and γ_m are their associated jump rates. In this section, I will not focus on the right-hand side of eq. (2.9), which is usually referred to as the Liouvillian. For the moment, I will just mention that solving eq. (2.9) is generally difficult because of the effective size of the density matrix, which is an object of size $|\mathcal{H}|^2$, where (recall) $\mathcal{H}$ is the Hilbert space of the main system of interest.

A rather common way of solving eq. (2.9) is the quantum trajectory approach, which takes advantage of the fact that a density matrix is an ensemble of vector states. This means, that a density matrix can be approximated by considering an ensemble of state vectors sampled from a probability distribution defined by the density matrix itself. At first, this remark seems circular, but it is not: the probability distribution can be sampled from simulating the "quantum jumps" induced by the jump operators, which act as external measurements onto the system. To this end, the GKSL equation can be re arranged as

$$\frac{d\rho}{dt} = -i(H_{\text{eff}}\rho - \rho H_{\text{eff}}^\dagger) + \sum_m A_m \rho A_m^\dagger, \tag{2.10}$$

where the effective non-Hermitian Hamiltonian is

$$H_{\text{eff}} = H - \frac{i}{2}\sum_m A_m^\dagger A_m. \tag{2.11}$$

Here we will explain a first-order approximation that gives rise to the quantum jump approach, although it is possible to perform higher-order approximations of it for numerical precision and stability. We follow the ideas by Daley [2014] next. Some time δt after t, the quantum state of our system will be

$$|\tilde{\psi}(t+\delta t)\rangle = (1 - iH_{\text{eff}}\delta t)\,|\psi(t)\rangle\,, \tag{2.12}$$

where the system has evolved an infinitesimal time under the action of the effective Hamiltonian. It is worth noting that the norm of this evolved state is not conserved because, in general, the effective Hamiltonian is non-Hermitian. The norm of the evolved state is $1-\delta p$, where

$$\delta p = \delta t \sum_m \langle\psi(t)|\, A_m^\dagger A_m\, |\psi(t)\rangle\,. \tag{2.13}$$

Now, let us pull the string from the other side. Within this framework of an infinitesimal time δt, we can see the GKSL equation as

$$\rho(t+\delta t) = \rho(t) - i\delta t(H_{\text{eff}}\rho(t) - \rho(t)H_{\text{eff}}^\dagger) + \delta t\sum_m A_m\rho(t)A_m^\dagger. \tag{2.14}$$

From eq. (2.12) one can check that

$$\begin{aligned}|\tilde{\psi}(t+\delta t)\rangle\langle\tilde{\psi}(t+\delta t)| &= (1 - iH_{\text{eff}})\,|\psi(t)\rangle\,\langle\psi(t)|\,(1 + iH_{\text{eff}}^\dagger)\\ &= |\psi(t)\rangle\langle\psi(t)| - i\delta t(H_{\text{eff}}\,|\psi(t)\rangle\langle\psi(t)| - |\psi(t)\rangle\langle\psi(t)|\,H_{\text{eff}}^\dagger),\end{aligned} \tag{2.15}$$

where the tilde indicates that the state does not satisfy the correct normalisation. Since the norm is $1-\delta p$, we can choose the state to be described by eq. (2.15) with probability $1-\delta p$, or with probability δp the state can be chosen to be "collapsed" to $A_m\,|\psi(t)\rangle$, where a particular collapse operator indexed by m is chosen with probability

$$\Pi_m := \delta t\,\langle\psi(t)|A_m^\dagger A_m|\psi(t)\rangle\,/\delta p. \tag{2.16}$$

As shown by Daley [2014], this results in an estimated state, whose average follows the GKSL equation, in the limit of infinite realisations, or infinite trajectories. Explicitly, from eq. (2.14) one can write [Daley, 2014]

$$\rho(t+\delta t) = (1-\delta p)\frac{|\tilde{\psi}(t+\delta t)\rangle\,\langle\tilde{\psi}(t+\delta t)|}{\sqrt{1-\delta p}\sqrt{1-\delta p}} + \delta p\sum_m A_m\,|\psi(t)\rangle\langle\psi(t)|\,A_m^\dagger, \tag{2.17}$$

for a pure state, but this is generalised to a mixed state by averaging over all possible trajectories.

In practice, at each δt one would decide if evolve or collapse the quantum state of the system. There are numerical methods that are more accurate, such as the work by Steinbach et al. [1995], where the time line is not discretised. Rather, the equation $\|\exp(-iH_{\text{eff}}t_1)\,|\psi(t_0)\rangle\|^2 = r$ is solved for a random number r between 0 and 1. Between time t_0 and t_1 the state is evolved under the effective Hamiltonian, and at time t_1 a quantum jump is performed, using one of the quantum jump operators.

We use this formalism to solve the GKSL equation that concerns our problem at hand:

$$\frac{d\rho}{dt} = i[\rho, H] + \kappa\mathcal{D}_b[\rho] + \gamma_b\sum_{j=1,2}\mathcal{D}_{\sigma_{vj}}[\rho] + \gamma_d\sum_{j=3,4}\mathcal{D}_{\sigma_{vj}}[\rho] + \gamma_\phi\sum_{j=1}^{4}\mathcal{D}_{\sigma_{jj}}[\rho], \tag{2.18}$$

where, for simplicity, we have used expressions for the dissipators $\mathcal{D}_A[\rho] = A\rho A^\dagger - \frac{1}{2}\rho A^\dagger A - \frac{1}{2}A^\dagger A\rho$. The four dissipative channels considered in eq. (2.18) are: phonon escape from the acoustic nanocavity at a rate κ with a collapse operator b due to unwanted coupling with leaky modes [Perea et al., 2004; Pascual Winter et al., 2007], spontaneous emission of the bright excitons at a rate γ_b with collapse operators σ_{vj} for $j = 1, 2$, effective spontaneous emission of the dark excitons (hole or electron spin flip followed by bright exciton spontaneous emission [Crooker et al., 2003]) at a rate γ_d with collapse operators σ_{vj} for $j = 3, 4$, and pure dephasing of all exciton states at a rate γ_ϕ with collapse operators σ_{jj} [Takagahara, 2002].

An example of a trajectory is shown in fig. 2.8. First, there is a dephasing jump

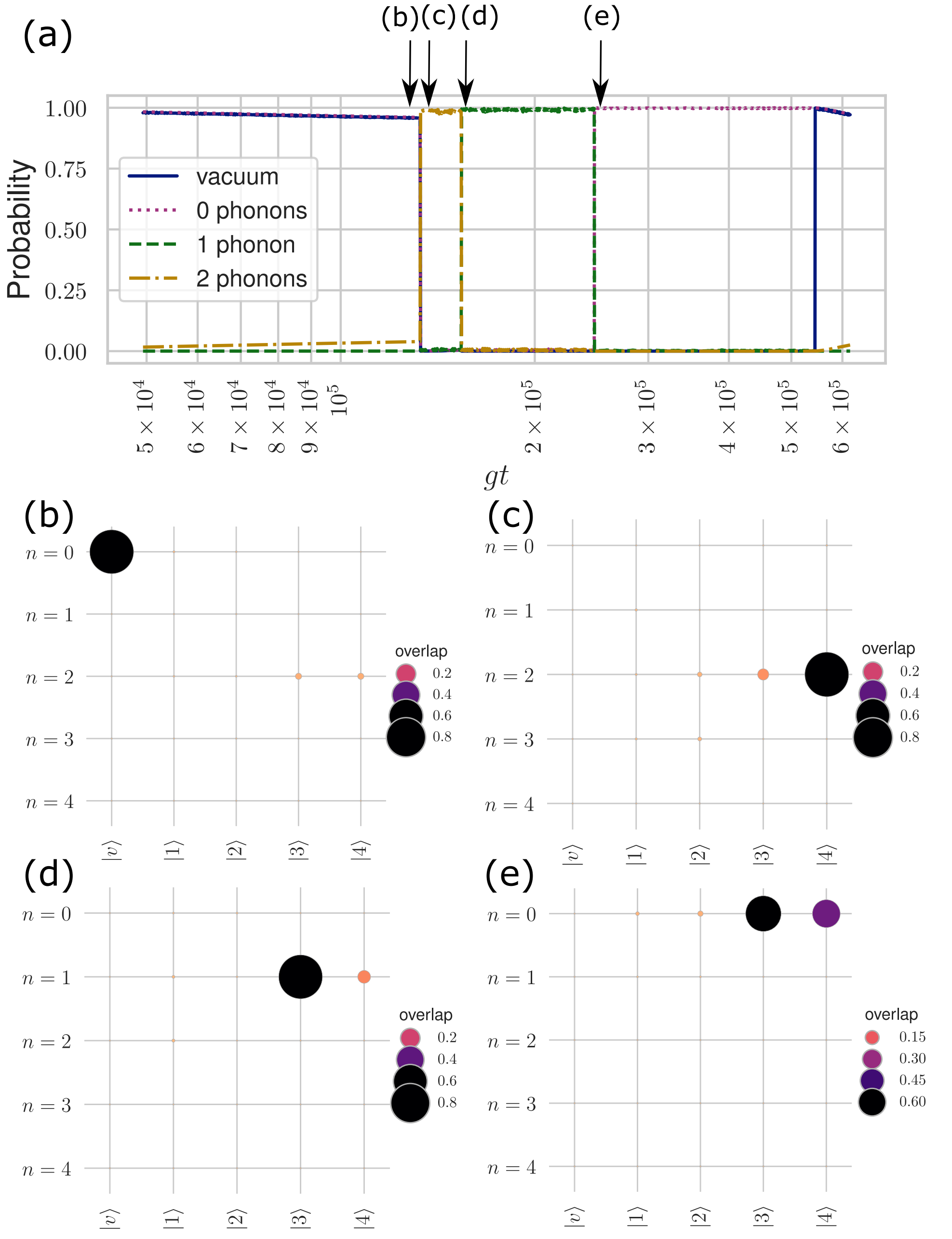

Figure 2.8: Evolution of a quantum trajectory due to Hamiltonian dynamics and dissipation-induced quantum jumps. (a) shows the occupation of the vacuum $|0, v\rangle$ state, and of the N-phonon states, i.e. the occupation of $\sum_i |N, i\rangle$. (b)-(d) show the state composition of the system at the times pointed out by an arrow in panel (a). Taken from Ref. [Vargas-Calderón et al., 2022].

that collapses the system into a state of two phonons and a dark state, as shown in fig. 2.8(c). Now, the phonon dissipative channel–with jump operator b–makes the cavity emit a phonon, leaving the system in a dark state, but with only one phonon, as shown in fig. 2.8(d). Another quantum jump occurs between the points (d) and (e) due to pure dephasing. This jump, however, does not affect the phonon population. Although the phonon-escape mechanism depletes the cavity phonon-by-phonon, it is shortly after the emission of the first phonon that the second phonon is also emitted due to the jump operator b, leaving the system in a dark excitonic state with zero phonons, as shown in fig. 2.8(e). This is a process of a two-phonon bundle emission. After the point (e), there are a couple of further dephasing quantum jumps that leave the phonon population unaffected because the exciton already occupies the dark states. Finally, the QD emits a photon due to an effective recombination of the dark exciton, i.e., a random spin flip followed by spontaneous emission, and the system is allowed to transit the same process once again. The two-phonon bundle emission process arises in all quantum trajectories, and becomes more common when the electron-phonon coupling g is increased, at the cost of realising giant Rabi oscillations between the vacuum and states that have a larger bright contribution.

2.4 CORRELATION FUNCTIONS AND “PHONOLUMINISCENCE” SPECTRUM

There are some very important quantities in quantum optics which characterise the kind of light that is emitted from dissipative processes. Of particular interest is the possibility to distinguish bunched light from anti-bunched light [Gerry et al., 2005]. The latter can only be achieved with a quantum source of light. The witness quantity that characterises this behaviour is a correlation function, namely the second-order correlation function of light

$$g^{(n)}(\tau = 0) = \langle b^{\dagger n} b^{n} \rangle \, / \, \langle b^{\dagger} b \rangle^{n} \tag{2.19}$$

for $n = 2$, which can be measured with the famous Hanbury Brown and Twiss interferometer [Brown and Twiss, 1956]. The second-order correlation function is related to the co-occurrence of single phonon detection events in this interferometer, as originally derived by Glauber [1963]. These correlation functions can also be used with the phonon field of the acoustic cavity, so instead of talking about properties of photon emission, we talk about properties of phonon emission through the dissipative channel of the acoustic cavity.

However, approaching multi-phonon physics through eq. (2.19) is inadecuate and insufficient because they are, in the experimental scenario, related to single-phonon events. Instead, we adopt the equal-time m-th order N-phonon bundle correlation function, defined as

$$g_{N}^{(m)}(\tau = 0) = \langle b^{\dagger N m} b^{N m} \rangle \, / \, \langle b^{\dagger N} b^{N} \rangle^{m} \, . \tag{2.20}$$

These generalised correlation functions correctly describe these multi-phonon properties because it treats the N-phonon bundle as a quasi-particle [Muñoz et al., 2014], with associated creation and annihilation operators $b^{\dagger N}$ and b^{N}, respectively.

In fig. 2.9 we show that this generalised correlation function reaches the sub-Poissonian anti-bunching regime for some values of the laser detuning [Zou and Mandel, 1990]. In this regime, it is possible to produce single isolated N-phonon bundles. The resonance associated to the dark symmetric with two phonons state presents the lowest value of $g_{2}^{(2)}$, which is located well within the anti-bunching regime. Analysis of the second order N-phonon bundle correlation function shows that the Bir-Pikus mechanism allows the generation of robust anti-bunching N-phonon bundles when the laser frequency is tuned to target giant Rabi oscillations

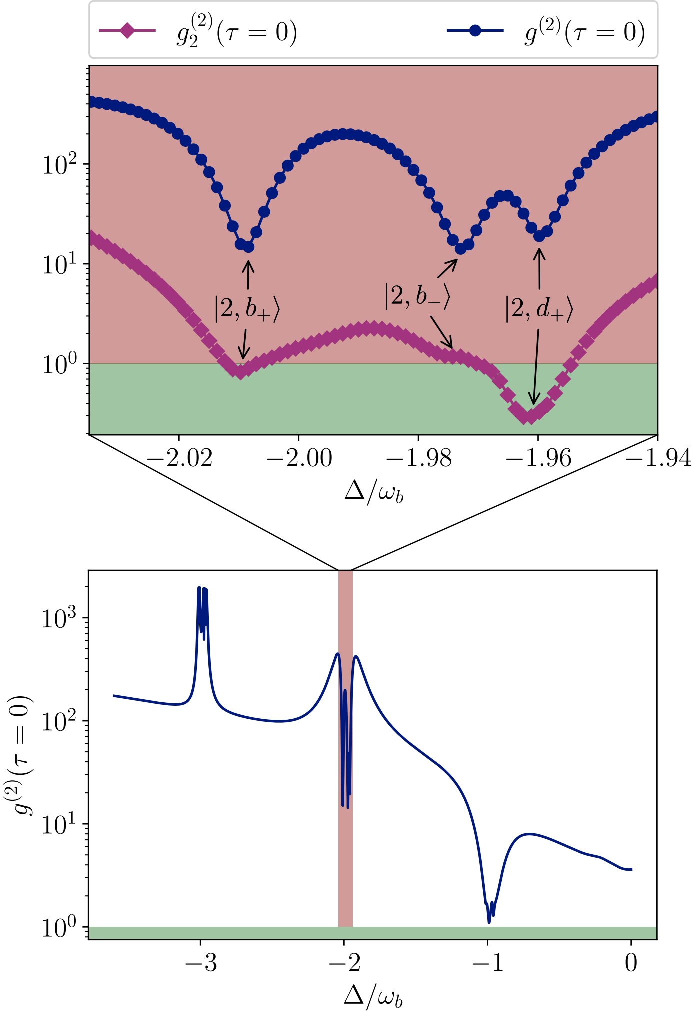

Figure 2.9: Equal-time one- and two-phonon bundle correlation functions. At $\Delta \approx -2\omega_b$ there are dips in the super-bunching peak of the usual correlation function at resonances corresponding to two-phonon states with bright (anti-)symmetric $|b_{(-)+}\rangle$, and with the dark symmetric state $|d_+\rangle$. The red/green-shaded regions align with sub-/super-Poissonian statistics regions. Taken from Ref.[Vargas-Calderón et al., 2022].

with the dark symmetric state. In fact, the values of the second-order N-phonon bundle correlation function are lower than those where the resonance involves the usual bright states.

Finally, we can take a look at the Wiener-Khintchine theorem [Mollow, 1969; Perea et al., 2004; Vargas-Calderón and Vinck-Posada, 2019], from which we calculate the "phonoluminiscence" spectrum, in analogy to the photoluminiscence spectrum

$$I(\omega) \propto \frac{\kappa}{\pi} \int_0^\infty \left\langle b(t) b^\dagger(t+\tau) \right\rangle e^{i\omega\tau}\, d\tau. \tag{2.21}$$

The phonoluminiscence spectrum tells us the intensity of emission of phonons at every different frequency. In particular, we see in fig. 2.10(c) such an spectrum with many different peaks. A more complete understanding of the Liouvillian is in place now.

We can interpret every single peak through the spectral theory of the Liouvillian superoperator $\mathcal{L}$ that satisfies eq. (2.18), written as $\frac{d\rho}{dt} = \mathcal{L}[\rho]$ [Petrosky and Pri-

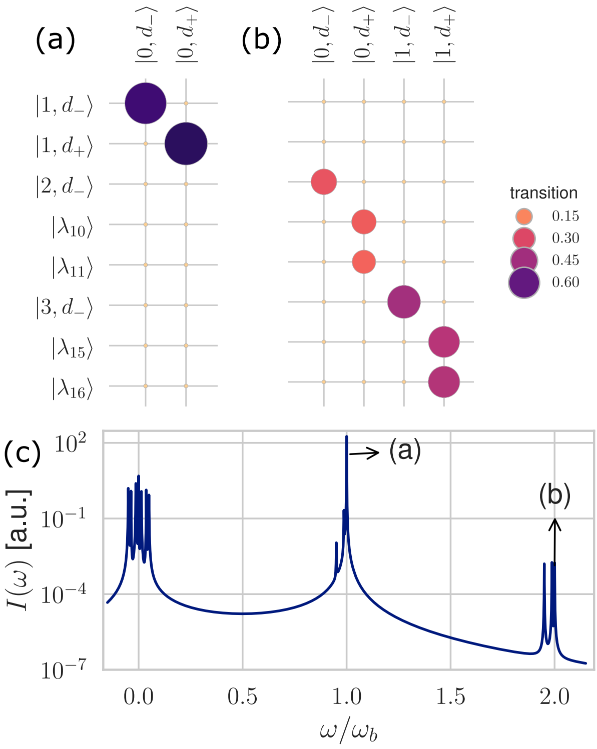

Figure 2.10: Phonon emission spectrum $I(\omega)$. Panels (a) and (b) show the elements $|\langle\psi|\,\varrho\,|\phi\rangle|^2$ of the Liouvillian eigenmatrices ϱ that match the peaks pointed in the phonon emission spectrum in panel (c). The eigenmatrices show that the peak (a) corresponds to a one phonon emission process, whereas the peak (b) corresponds to two-phonon emission processes. Eigenstates $|\lambda_{10}\rangle$ and $|\lambda_{11}\rangle$ of eq. (2.3) are superpositions of $|0,v\rangle$ and $|2,d_+\rangle$, and eigenstates $|\lambda_{15}\rangle$ and $|\lambda_{16}\rangle$ are superpositions of $|0,v\rangle$ and $|3,d_+\rangle$. Taken from Ref. [Vargas-Calderón et al., 2022].

gogine, 1996; Petrosky, 2010; Manzano and Hurtado, 2018]. A formal solution of this equation for a time-independent Liouvillian is

$$\rho(t) = \sum_k e^{\Lambda_k t}\,\mathrm{Tr}[\varrho_k \rho(0)]\varrho_k, \tag{2.22}$$

where Λ_k are the complex eigenvalues of $\mathcal{L}$ with corresponding eigenmatrices ϱ_k. The eigenvalues are associated to the emission peaks [Albert and Jiang, 2014; Hatano and Petrosky, 2015; Vargas-Calderón and Vinck-Posada, 2020] as they show both the peak location $\mathrm{Im}\{\Lambda_k\}$ and the full width at half maximum $-\mathrm{Re}\{\Lambda_k\}$. On the other hand, the eigenmatrices account for information about which states are involved in each transition [Tay and Petrosky, 2008]. The examination of the eigenmatrices matrix elements still shows which states are involved in the transitions, as displayed in fig. 2.10(a) and (b).

Figure 2.10(a) shows that the most prominent peak of the spectrum matches transitions between dark states with one phonon and dark states with zero phonons. In Ref. [Vargas-Calderón et al., 2022], we highlight the fact that the magnitude of the matrix elements shown in fig. 2.10(a) and (b) does not indicate the contribution of those transitions to the emission peak. Also, fig. 2.10(b) shows the allowed transitions for another peak, at a frequency $\omega = 2\omega_b$, which matches transitions between dark states with two phonons decaying to dark states with no phonons, and also dark states with three phonons decaying to dark states with one phonon. Here, the emission peak coinciding with the two-phonon emission is significantly smaller than the one-phonon emission peak. However, the corresponding transitions can be differentiated due to the large energy difference in the spectrum.

Again, we refer the reader to Ref. [Vargas-Calderón et al., 2022] for a complete account of the parameters used in this work, as well as a discussion of experimental feasibility given realistic parameters.

2.5 WHAT IS THE RELEVANCE OF GIANT RABI OSCILLATIONS FOR QUANTUM INFORMATION PROCESSING?

So far, we have promoted the study of these sort of "fundamental" systems because they serve as building blocks for quantum technologies that enable quantum information processing. In this short section, we review the main ideas that allow us to assert this; thus, putting our work in synchrony with the ultimate goal of enabling (quantum) computational techniques to study quantum physics.

The principle under which current quantum computing technologies work is by controlling the time during which a system undergoes Rabi oscillations. The Rabi oscillations are induced by external electric and magnetic fields, applied during some finite time. In what follows, I will give a short primer on controlling the quantum state of this such system. However, the relevance of our work is that the following procedure can also be used to produce giant Rabi oscillations between states composed of N-phonon bundles. By varying the targeted N-phonon state, we can promote the usual two-level system state to a qudit, or d-level system. Thus, our work is useful for controlling the state of a system that is higher-dimensional than a simple qubit. As we will see in section 5.2, having qudits instead of qubits renders extremely useful for solving important problems with quantum computers.

We start off by discussing how the interaction between a classical field and a two-level system is considered into a Hamiltonian such as eq. (2.5). This is the main ingredient for quantum control: since we are able to cherry-pick the N-phonon state that we want to couple to create giant Rabi oscillations, we can use the same working principle of controlling a two-level system to perform operations with the chosen N-phonon states. Therefore, we introduce the main tools to understand these two-level system quantum control techniques, which require using external fields to control the state of the two-level system.

2.5.1 The Bloch sphere

Usually, the Bloch sphere is introduced via the possible states a qubit (a two-level system) which is characterised by the so-called Bloch sphere angles θ and φ:

$$|\psi\rangle = \cos\frac{\theta}{2}\,|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}\,|1\rangle\,. \tag{2.23}$$

The state can be visualised in the Bloch sphere, shown in fig. 2.11.

In our case, it is best to introduce the Feynman-Vernon-Hellwarth representation of the qubit as a precessing spin [Feynman et al., 1957]. Let us do so by introducing the Hamiltonian of a two-level system driven by an external laser. As before, in the rotating frame of the laser at frequency ω, the Hamiltonian simply reads

$$\Delta\sigma^\dagger\sigma + \frac{\Omega}{2}(\sigma^\dagger + \sigma). \tag{2.24}$$

We consider the general state $\tilde{\rho}$ of the two-level system

$$\tilde{\rho} = \begin{bmatrix} \tilde{\rho}_{ee} & \tilde{\rho}_{eg} \\ \tilde{\rho}_{ge} & \tilde{\rho}_{gg} \end{bmatrix} \tag{2.25}$$

where the tildes indicate that populations of the excited $\tilde{\rho}_{ee}$ and ground $\tilde{\rho}_{gg}$ states, as well as the coherences $\tilde{\rho}_{ge}$ and $\tilde{\rho}_{eg}$ are in the rotating frame of the laser. Populations

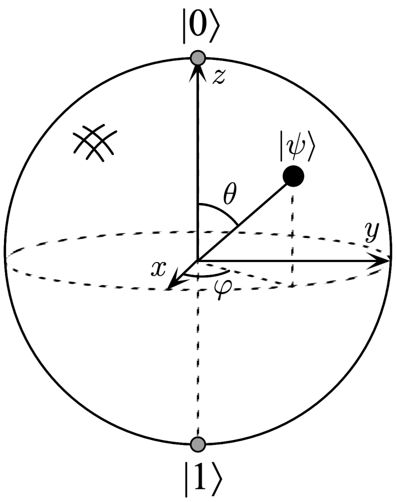

Figure 2.11: Bloch sphere representation of the state of a two-level system described by eq. (2.23). Taken from Ref. [Nielsen and Chuang, 2002].

do not change in any frame: $\rho_{ee} = \tilde{\rho}_{ee}$ and $\rho_{gg} = \tilde{\rho}_{gg}$, but coherences rotate: $\rho_{ge} = \tilde{\rho}_{ge} e^{-i\omega t}$ and $\rho_{eg} = \tilde{\rho}_{eg} e^{i\omega t}$.

It can be verified [Steck, 2007] that the following equation holds:

$$\frac{\mathrm{d}\langle\sigma\rangle}{\mathrm{d}t} = \mathcal{P} \times \langle\sigma\rangle\,, \tag{2.26}$$

where $\langle\sigma\rangle = (\langle\sigma_x\rangle, \langle\sigma_y\rangle, \langle\sigma_y\rangle)^T = (\tilde{\rho}_{eg} + \tilde{\rho}_{ge}, i(\tilde{\rho}_{eg} - \tilde{\rho}_{ge}), \rho_{ee} - \rho_{gg})^T$ and $\mathcal{P} = (\Omega, 0, \Delta)^T$. Note that this equation is very similar to the angular momentum equation $\frac{\mathrm{d}\boldsymbol{L}}{\mathrm{d}t} = \boldsymbol{\omega} \times \boldsymbol{L}$, where $\boldsymbol{L}$ is the angular momentum vector and $\boldsymbol{\omega}$ is the angular frequency vector.

Remarkably, since the rate of change of the vector $\langle\sigma\rangle$, which we now call the Bloch vector [Bloch, 1946], is perpendicular to $\langle\sigma\rangle$ itself, the norm of $\langle\sigma\rangle$ will remain constant. Therefore, if we have a pure state, it can be easily checked that the norm of this vector is one, meaning that the Bloch vector in either the rotating frame of the laser or in the canonical frame will be restricted to te surface of a sphere: the Bloch sphere in fig. 2.11.

We can immediately see that the vector $\mathcal{P}$ is extremely important because it is the precession vector that dictates the trajectory that the Bloch vector follows on the surface of the Bloch sphere. In particular, if the laser is at resonance with the two-level system, $\Delta = 0$, and in this condition, $\mathcal{P}$ points to the x axis, and the states will follow circles perpendicular to the x axis, as shown in fig. 2.12.

The velocity of the rotation is related to the magnitude of the precession vector $\sqrt{\Omega^2 + \Delta^2}$, which is the so-called Rabi frequency. You can start to see how controlling the state of the two-level system works: you turn on and off lasers with detunings and powers that can be controlled with piezoelectrics and voltages, respectively, which allows you to control the direction and magnitude of the precession vector. Thus, manipulation of the state of the two-level system follows.

An important final remark is that finite pulses of lasers, unfortunately, do not have a single frequency, as we have so far assumed. As you may know, the Fourier transform of a time signal will give you its frequency spectrum. Very short signals have a wide spectrum. What is the problem with this? For starters, the simplified Hamiltonian in eq. (2.24) is not a physical Hamiltonian: it is an approximation of an usually much more complicated Hamiltonian (say, a quantum harmonic oscillator). The more complicated Hamiltonian does not have two levels, but more (even infinite), and having a short laser pulse means that some frequencies might excite levels that are not of our interest. Pulse design is an entire area of research, which has also received significant attention from the symbiosis between machine learning

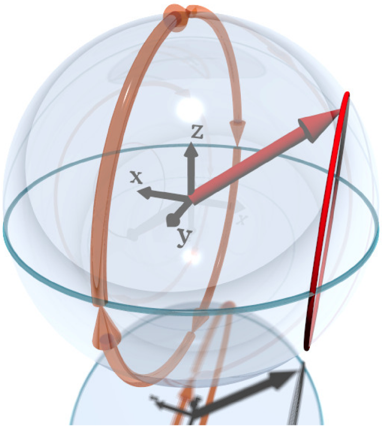

Figure 2.12: Trajectories of a state initially in the ground state (orange path) or in a superposition state (red path), when the laser is in resonance with the two-level system, so that the precession vector $\mathcal{P}$ points in the x direction.

and quantum physics [Zahedinejad et al., 2016; Yang et al., 2018; Ding et al., 2019; Dong et al., 2016]. Indeed, pulse design is a very interesting tool to suppress non desirable resonances so that non-intended levels are never excited (see for instance the derivative removal by adiabatic gate method [Motzoi et al., 2009; Gambetta et al., 2011]).

2.6 SUMMARY AND OUTLOOK

The take home message here is that our work gives a path way to couple two states of a multi-level system through giant Rabi oscillations, which can be used to effectively treat the system as a two-level system. Here, we took advantage of a phonon cavity, which is difficult to control with current technology, but remains an interesting platform as phonon control technology advances. Since said states involve dark-exciton states, they are much more robust to spontaneous emission (recombination). Combining giant Rabi oscillations with the usual quantum control techniques can yield a toolbox for high-level (qudit) quantum computing. As we later show in this thesis, qudit systems can be very beneficial for combinatorial problems (see section 5.2), and, of course, it goes without saying that the possibility of high-level quantum computing is an incredible asset per se, as we also show that quantum neural networks might perform better than neural networks for quantum problems, as we show in section 3.4.

Also, as we show in section 5.2, it is of the uttermost interest the ability to control how quantum systems interact to be able to simulate different sorts of Hamiltonians that not only allow an analogue (or even digital) processing of quantum information, but also allow finding ground states through, for instance, quantum annealing of such interaction-engineered systems.

3 NEURAL QUANTUM STATES AND PHASE TRANSITIONS

As mentioned in the introduction (cf. fig. 1.1), the space of pictures is huge. However, the space of images of memes, or human faces as in fig. 1.2, is much more restricted: it is a manifold within the image space. Think about it... if the skin of a person is of a given colour, all the pixels in this person's cheeks will almost be of the same colour: there is a restriction on the values that pixels can take because they describe an object that, most of the times, changes its colour smoothly with respect to the position in the image. Something similar happens with many quantum physics problems, where ground states or steady states tend to follow a particular structure. In this chapter we will tackle three different problems: finding the ground state of a Hamiltonian, finding the steady state of a GKSL equation (see eq. (2.9)), and finding the time-evolution of a system described by a Hamiltonian. All of these problems require to find a probability distribution defined over the Hilbert space, usually in the canonical basis. This is analogous to finding the manifold in the image example. In machine learning this is called, as we mentioned in chapter 1, the curse of dimensionality. In quantum physics we refer to these problems as intractable, due to the usually large Hilbert spaces involved. Our approach to these problems will be neural quantum states or neural network ansätze [Carleo and Troyer, 2017].

With Herbert we were concerned with the limitations of using neural networks as ansätze of the variational method (in its Monte Carlo flavour, of course), and started to test the method with some models using systematic experiments. Along with Fabio, we published another set of papers on these experiments, which we will expose in this chapter.

Our work has opened a line of research mainly in the Superconductivity and Nanotechnology group, which is the line of artificial intelligence methods for quantum physics (probably the first of its kind in Colombia). This is probably the legacy of this thesis project that has been closely accompanied by Herbert and Fabio[1]. As of now, our group has a PhD student who is working on machine learning techniques for the preparation of states in qudit systems, and a masters student who is working on an ansatz so far overlooked by the physics + machine learning community (maybe because neural networks attract so much attention): gradient boosting to represent ground states (and steady states in the future). There are also three undergraduate students who work on open quantum system dynamics with quantum computers, kernel methods in quantum computing for supervised and unsupervised machine learning applications, and simulation of quantum circuits with forests of decision trees.

In this chapter, an overview of the state of the art of neural quantum states will be given in section 3.1. Then, in section 3.2, the NetKet project–an open source platform for many-body physics with neural networks–will be presented. Finally, applications to ground states, dynamics and steady states will be explored.

[1] I should say that both Herbert and Fabio have exploited our collaboration in their respective groups to open new lines of research. In the case of Fabio's group there has been enormous work with many students in the direction explored in chapter 4.

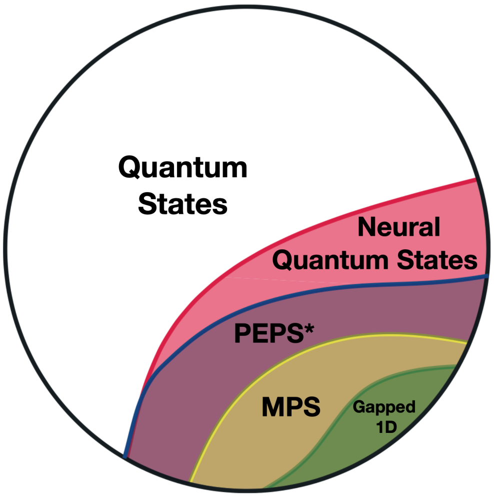

Figure 3.1: Expressive power of classically tractable variational quantum states. Figure taken from Ref. [Sharir et al., 2021]

3.1 STATE OF THE ART

There is an inherent difficulty in quantum mechanics, but people have demonstrated to be very recursive to handle intractable Hilbert spaces, by either taking advantage of the Hamiltonian properties (usually its symmetries), or just by coming up with clever ways of exploring a Hilbert space. As I mentioned in chapter 1, I got close to one of such clever ways of exploring a large Hilbert space when I read and reproduced the work by Saito [2017]. The method that was used in this work is called variational Monte Carlo (I will talk about this method in detail in this chapter), which leverages Monte Carlo methods to perform the variational method of quantum mechanics. The variational method is a widely known method both in physics education and research, and is introduced in early quantum mechanics courses [Ballentine, 2014; Le Bellac, 2011]: a variational wave function (an ansatz) has some parameters, and we minimise the expected energy with respect to such variational wave function by calculating the gradients of this expected value with respect to the variational parameters. A simple idea, a powerful idea, the idea of optimising a lower-bounded function!

Modern research coarsely focuses on three "novel" ways to exploit variational methods: tensor networks [Cirac et al., 2021], variational circuits [Cerezo et al., 2021] and neural quantum states [Carleo et al., 2019b; Carrasquilla, 2020; Vivas et al., 2022]. State of the art results, however, are achieved only through tensor network approaches and neural quantum states, because variational circuits have to be classically simulated, limiting the size of the problems that can be tackled. Tensor networks are much more mature, and have been increasingly improving with the availability of faster and larger classical computers. Moreover, there is a rather good understanding of the most common tensor network architectures, which helps in proving bounds for approximating ground states and observables [Zhou et al., 2008; Huang, 2019]. Neural quantum states, on the other hand, are much younger, but are showing promising results [Roth and MacDonald, 2021; Choo et al., 2019; Chen et al., 2022; Vieijra and Nys, 2021], and are hypothesised to be able to reproduce tensor network states [Sharir et al., 2021] as shown in fig. 3.1. However, their capabilities are not that well understood, which is a limitation inherited from the deep learning area within machine learning: how neural networks learn is not well understood yet.

In this chapter we focus on neural quantum states, which were introduced in the work by Carleo and Troyer [2017]. Three years ago, I found a fantastic Python library called NetKet[2] which allowed researchers to quickly prototype different neural network ansätze and play around minimising the expected energies of different Hamiltonians. I became an official collaborator of NetKet, and we eventually published a paper on NetKet 3 [Vicentini et al., 2021], which is the Python-only version of NetKet[3]. This library is one of the central pieces of work of this thesis. For now, let us introduce some concepts central to neural quantum states, and by extension, central to NetKet. The rest of this short state of the art presents the variational Monte Carlo method in section 3.1.1 and the so-called neural quantum states, or neural network ansätze in section 3.1.2.

3.1.1 Variational Monte Carlo

Variational Monte Carlo is a method that allows the controlled exploration of an intractable Hilbert space in order to carry out the variational method of quantum mechanics. The variational method comes from the fact that the expected value of a Hamiltonian with respect to any quantum state is always less or equal than the ground state energy (it is only equal when said state is the ground state). Thus, the variational condition is usually written as [Ballentine, 2014]

$$\langle \psi | H | \psi \rangle = \langle H \rangle_\psi \geq E_0 \tag{3.1}$$

where E_0, also sometimes written E_g, is the ground state energy.

Now, the variational method requires a variational wave function, which is just a parameterised quantum state. It can be written as $|\psi_{\boldsymbol{\theta}}\rangle$, where $\boldsymbol{\theta}$ is just a vector of parameters. Let us denote a canonical orthonormal basis of the Hilbert space by $\{|\boldsymbol{b}\rangle\}$, where $|\boldsymbol{b}\rangle$ serves as a place holder for an element of that basis. If we simply write $\boldsymbol{b}$, we are referring to a classical representation of a vector with the entries of that element. This place holder is called a configuration of the system. To give a concrete example, if the Hilbert space is that of 100 spins, then the spins can be either up or down in this canonical basis. If we denote spins up by 1 and spins down by 0, $\boldsymbol{b}$ are just bit strings of length 100. Since the basis is orthonormal, we have that for every pair of different configurations of the Hilbert space $|\boldsymbol{b}\rangle, |\boldsymbol{b}'\rangle \in \mathcal{H}$

$$\langle \boldsymbol{b} | \boldsymbol{b}' \rangle = \prod_i \delta_{b_i, b_i'}, \tag{3.2}$$

where I have used the Kronecker delta.

With such a canonical basis, we can write the variational state $|\psi_{\boldsymbol{\theta}}\rangle$ with the help of the variational wave function $\psi_{\boldsymbol{\theta}} : \mathcal{H} \to \mathbb{C}$ as follows:

$$|\psi_{\boldsymbol{\theta}}\rangle = \sum_{\boldsymbol{b}} \psi_{\boldsymbol{\theta}}(\boldsymbol{b}) |\boldsymbol{b}\rangle, \tag{3.3}$$

which results from multiplying the variational state by the identity written in the given basis, so that $\psi_{\boldsymbol{\theta}}(\boldsymbol{b}) = \langle \boldsymbol{b} | \psi_{\boldsymbol{\theta}} \rangle$. This allows us to write the expectation value of an operator A as [Becca and Sorella, 2017]

$$\langle \psi_{\boldsymbol{\theta}} | A | \psi_{\boldsymbol{\theta}} \rangle = \sum_{\boldsymbol{b}, \boldsymbol{b}'} P_{\boldsymbol{\theta}}(\boldsymbol{b}) \langle \boldsymbol{b} | A | \boldsymbol{b}' \rangle \frac{\psi_{\boldsymbol{\theta}}(\boldsymbol{b}')}{\psi_{\boldsymbol{\theta}}(\boldsymbol{b})}, \tag{3.4}$$

where $P_{\boldsymbol{\theta}}(\boldsymbol{b}) = |\psi_{\boldsymbol{\theta}}(\boldsymbol{b})|^2 / \sum_{\boldsymbol{b}'} |\psi_{\boldsymbol{\theta}}(\boldsymbol{b}')|^2$ is the probability of the configuration $\boldsymbol{b}$ for the induced probability distribution given by the variational state. You can see that

[2] By the time, the library was written in C++, but there was a plan to use machine learning frameworks written in Python, such as Pytorch from Facebook (now Meta), to make development of the library more amenable. The parts that did not involve directly neural network operations were handled by numba's just-in-time compilation, which leveraged numpy code to C++ under the hood, making Python code blazing fast. I happened to discover this library right on this transition, and I decided to learn software development and quantum physics by involving myself in helping the NetKet team.

[3] The team moved to Google's JAX framework, which is based on functional programming in Python

computing an expected value of some operator is very difficult: one needs to iterate over all the elements of the basis $\boldsymbol{b}$, and for each of them one needs to compute some ratio of wave functions for every other element of the basis that is connected to $\boldsymbol{b}$. Running through all the elements of the basis is prohibitive because they are usually intractable. The second sum, however, is not that problematic. Usually the operator A will be the Hamiltonian, which tends to be sparse in the canonical basis. Actually, this happens for almost every quantum operator that one handles on a daily-basis because of the locality of interactions: particles that are far away from each other do not directly interact (effectively). Sparsity mainly says that if you fix $\boldsymbol{b}$, then $\langle \boldsymbol{b}| A |\boldsymbol{b}'\rangle$ will be almost always zero, except from some small set of $\boldsymbol{b}'$s, which are called the connected elements of $\boldsymbol{b}$ with respect to A.

To our rescue comes the probability term in eq. (3.4). Basically, it says that if, somehow, we are able to efficiently build a sample $\mathcal{M}$ distributed according to $P_{\boldsymbol{\theta}}(\boldsymbol{b})$, we can approximate eq. (3.4) with

$$A^{\star}_{\boldsymbol{\theta}} = \left\langle \sum_{\boldsymbol{b}'} \langle \boldsymbol{b}| A |\boldsymbol{b}'\rangle \frac{\psi_{\boldsymbol{\theta}}(\boldsymbol{b}')}{\psi_{\boldsymbol{\theta}}(\boldsymbol{b})} \right\rangle_{\boldsymbol{b}\in\mathcal{M}}, \tag{3.5}$$

where $A^{\star}_{\boldsymbol{\theta}}$ is just notation for an estimated value of the expected value of A with respect to the variational state $|\psi_{\boldsymbol{\theta}}\rangle$.

In particular, we can use Monte Carlo methods to sample from $P_{\boldsymbol{\theta}}(\boldsymbol{b})$ in order to get our desired sample of configurations $\mathcal{M}$. Again, note that the sum in eq. (3.5) is tractable, as most of the terms in the sum are just zero because of the assumed sparsity of the Hamiltonian. Apart from this sparsity which we usually encounter in Hamiltonians, there is a remarkable gift from nature: usually (YES, USUALLY! but not always, as we show in section 3.4), ground states lie in a small corner of the Hilbert space, meaning that the size of $\mathcal{M}$ need not be large: only a few number of states is necessary to accurately describe the ground state of a Hamiltonian. The problem, which is very challenging, is to find which states those might be!

A common Monte Carlo method for creating the sample is the Metropolis-Hastings algorithm [Hastings, 1970], which goes like this. Unless you have any intuition about the Hamiltonian (I will replace A with H from now on because our main interest will be the Hamiltonian operator), you can randomly generate a configuration $\boldsymbol{b}^{(0)}$, where the (0) just denotes that we are at iteration 0. You need a detailed balance-like probabilistic update rule, which is a probability map g from which you can compute $g(\boldsymbol{b}'|\boldsymbol{b})$, namely, the probability that $\boldsymbol{b}$ gets updated to $\boldsymbol{b}'$. The detailed balance property is explicitly $g(\boldsymbol{b}'|\boldsymbol{b}) = g(\boldsymbol{b}|\boldsymbol{b}')$. At iteration t you do as follows.

- With the rule g you will sample a candidate sample $\tilde{\boldsymbol{b}}^{(t)}$ from the previously known configuration $\boldsymbol{b}^{(t-1)}$.
- You will set $\boldsymbol{b}^{(t)}$ to be the candidate $\tilde{\boldsymbol{b}}^{(t)}$ with probability

$$\min\left\{1, \frac{|\psi_{\boldsymbol{\theta}}(\tilde{\boldsymbol{b}}^{(t)})|^2}{|\psi_{\boldsymbol{\theta}}(\boldsymbol{b}^{(t-1)})|^2}\right\}, \tag{3.6}$$

  otherwise you set $\boldsymbol{b}^{(t)} := \boldsymbol{b}^{(t-1)}$.

By collecting configurations in this fashion, one is guaranteed to sample exactly from $P_{\boldsymbol{\theta}}(\boldsymbol{b})$. Therefore, we are able to efficiently compute expectation values of operators in intractable Hilbert spaces. All that is left is to minimise the expected value, which can be done with gradient-based methods widely used in machine learning, through either stochastic gradient descent or stochastic reconfiguration [Becca and Sorella, 2017]. Iteratively, the variational energy in eq. (3.5) is minimised, producing a new set of parameters $\boldsymbol{\theta}$. This process resembles an Expectation Maximisation algorithm [Dempster et al., 1977], or also a reinforcement learning algorithm where

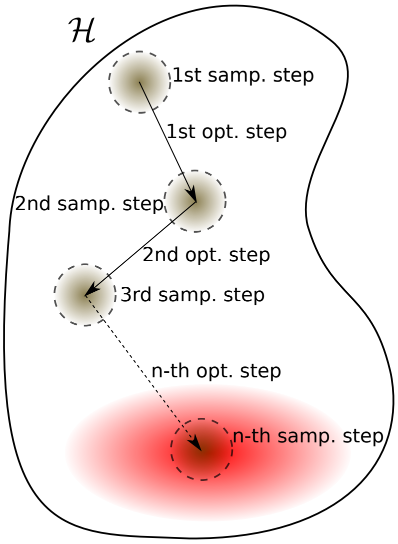

Figure 3.2: Representation of the variational Monte Carlo technique. With randomly initialised parameters $\boldsymbol{\theta}$, a set of states from the Hilbert space $\mathcal{H}$ is sampled. By minimising the energy defined in eq. (3.5), the parameters $\boldsymbol{\theta}$ are updated. These two steps are repeated a number of times with the objective of sampling the states (in the occupation basis) that are relevant for the ground state, depicted by a blot at the bottom, with the correct probability distribution. Figure taken from Ref. [Vargas-Calderón et al., 2020].

the agent is the Monte Carlo algorithm in charge of taking actions (picking new configurations to form the sample) based on its brain which captures information about the environment (the variational wave function that tells it where the ground state might be). A schematic representation of the variational Monte Carlo technique is shown in fig. 3.2.

The red blot at the bottom of fig. 3.2 depicts a region of the Hilbert space that contains the most relevant states for the ground state of whatever Hamiltonian is being studied. The Markov chain formed by the Metropolis-Hastings algorithm is also portrayed here as a way to wander about the Hilbert space in search for this red blot. In that regard, the region of relevant states can contain a variable number of states depending on the parameters of the Hamiltonian that one is intending to solve. Therefore, an issue immediately arises: for an unknown target probability distribution $|\psi(\boldsymbol{b})|^2$, the sampling can be too small or too large with respect to the size of the region of relevant states. If it is small, important information about the ground state might not be taken into account, whereas if it is large, noisy probability from non-relevant states can be taken into account. This is unfortunate, but that is what we have (and it works pretty well up to a point, as we show in this chapter).

3.1.2 Neural Quantum States

Neural quantum states were introduced by Carleo and Troyer [2017]. The name "neural" quantum state comes from the fact that the variational wave function in eq. (3.3), with which the variational energy estimation is computed in eq. (3.5), is a neural network. In particular, Carleo and Troyer [2017] used a restricted Botlzmann machine. From now on, I will present the neural networks used as variational wave functions in this work for closed quantum systems. In section 3.5 we will see that we need a variational density matrix in order to model an open quantum system, and the corresponding Deep Boltzmann Machine will be introduced there.

Restricted Boltzmann Machine

We are interested in problems where bosons occupy sites on a lattice (note that the lattice might be a real space lattice, or a momentum lattice too). This means that the basis configurations $\boldsymbol{b}$ that we introduced previously are occupation vectors of the form $(n_1, n_2, \ldots)$, where n_i is the occupation (number of bosons) in the i-th site of the lattice. Note that the vector can be infinitely long. In this Fock space basis, the ground state of a boson model can be written as

$$|\psi\rangle = \sum_{n_1=0, n_2=0, \ldots}^{\infty} \psi(n_1, n_2, \ldots) \, |n_1, n_2, \ldots\rangle \, , \tag{3.7}$$

where $|\psi(n_1, n_2, \ldots)|^2$ are the probability amplitudes corresponding to the Fock states $|n_1, n_2, \ldots\rangle$. In order to map the wave function to a computer, both the number of sites and the number of possible particles in each site have to be truncated. We will refer to the number of sites as N and to the maximum number of particles in each site as $M - 1$. The coefficients $\psi(n_1, n_2, \ldots, n_N)$ are approximated by the restricted Boltzmann machine (RBM).

RBMs are generative neural networks, formally described by a bipartite undirected graph such as the one shown in fig. 3.3, where there is a layer of visible neurons denoted by $\boldsymbol{v}$ that are used to input real data, and a layer of hidden neurons denoted by $\boldsymbol{h}$ that are used as latent variables of the model [Smolensky, 1986].

In particular, the wave function coefficients take the form

$$\psi(n_1, n_2, \ldots, n_N) \approx \psi_{\boldsymbol{\theta}}(n_1, \ldots, n_N) = \sum_{\boldsymbol{h}} \exp(-E_{\text{RBM}}(\boldsymbol{v}(\boldsymbol{n}), \boldsymbol{h})), \tag{3.8}$$

where $E_{\text{RBM}}(\boldsymbol{v}(\boldsymbol{n}), \boldsymbol{h})$ is the energy of the RBM [Smolensky, 1986], and $\boldsymbol{\theta}$ are the variational parameters of the RBM (I will specify these in a bit). As a short-hand notation, an occupation configuration is denoted as $\boldsymbol{n}$ (yes, $\boldsymbol{n}$ is a bit more intuitive than the previously used $\boldsymbol{b}$ for this particular case), and it is inputted to the visible layer of the RBM. However, the configuration first needs to be one-hot encoded as follows: each occupation n_i is encoded into an M-component vector whose j-th component is δ_{j,n_i}, $j = 0, 1, \ldots, M-1$; then, the vectors for every occupation are concatenated into $\boldsymbol{v}(\boldsymbol{n})$. In other words, the occupation of each site, say n_i can take M values, but the RBM is made to accept binary values only. Therefore, we convert each occupation n_i into a vector of length M for which all of its entries are -1 except for the n_i-th one, which is 1.

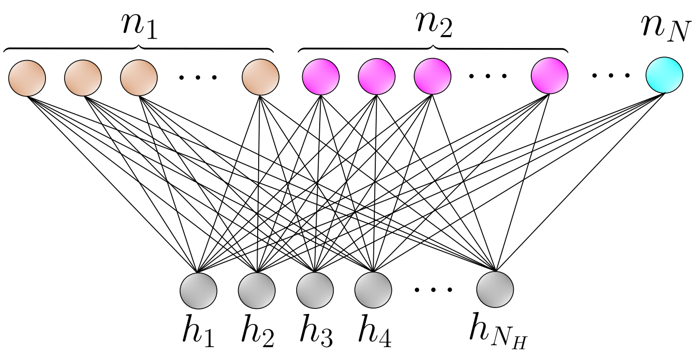

Figure 3.3: Illustration of the RBM, where each site occupation is one-hot encoded into M visible neurons, depicted with different colours for different sites in the top layer. There are N_H hidden neurons in the bottom layer connected with the visible neurons through weights $W_{\ell,j}$ that connect the ℓ-th hidden neuron with the j-th visible neuron. Taken from Ref. [Vargas-Calderón et al., 2020].

Moreover, if the N_H hidden neurons are restricted to binary values 1 or -1, then, the approximated wave function coefficients can be written as [Carleo and Troyer, 2017]

$$\psi_{\boldsymbol{\theta}}(\boldsymbol{n}) = e^{\sum_j a_j v_j(\boldsymbol{n})} \prod_{\ell=1}^{N_H} 2\cosh\left(b_\ell + \sum_j W_{\ell,j} v_j(\boldsymbol{n})\right), \tag{3.9}$$

where a_j, b_ℓ and W are the complex-valued visible bias, hidden bias and connection matrix of the RBM, respectively.

Thus, the approximation of the wave function coefficients is done by adjusting the parameters $\boldsymbol{\theta} : \{a_j, b_\ell, W_{\ell,j}\}$ that minimise the energy $\langle \psi_{\boldsymbol{\theta}} | H | \psi_{\boldsymbol{\theta}} \rangle$ for a given boson Hamiltonian H.

Auto-regressive Neural Networks

Sampling in variational Monte Carlo can be a big problem because we need algorithms that quickly converge to the desired probability distribution. Since we do not know the desired probability distribution, measuring whether we are converging to it is impossible (though there are metrics for convergence of Markov chains to some distribution [Roy, 2020], which can be a stationary probability distribution and not the desired probability distribution *per se*). In practice, we know that the Metropolis-Hastings algorithm that we described before can become stuck at stationary probability distributions which do not thermalise to the desired probability distribution. Before discussing a possible remedy (auto-regressive sampling) for this possible upset, let us discuss an interesting figure (fig. 3.4) from Vivas et al. [2022], which shows a principal component analysis plot (this is a technique of dimensionality reduction, which projects data in a high-dimensional space onto a lower-dimensional space–2D in this case– by means of a linear map that maximises the variance of the data on the lower-dimensional space) of data sampled using the usual Markov chain Monte Carlo methods versus auto-regressive sampling.

Figure 3.4 gives a clear picture of what I am referring to. Sampling from Markov chains inevitably leads to sampled configurations being correlated to one another. This is a simple result because in order to get a new configuration, one applies probabilistic rules to get the next configuration (Hamiltonian Markov chain Monte Carlo is just another way of computing the acceptance probability, but falls within the framework of the Metropolis-Hastings algorithm that I have explained previously). Therefore, the next configuration is correlated to the previous one. On the other hand, as we will see, auto-regressive models sample exactly from $P_{\boldsymbol{\theta}}(\boldsymbol{b})$, which means that configurations are completely uncorrelated with one another, because sampling a configuration does not require a seed configuration to begin with. This

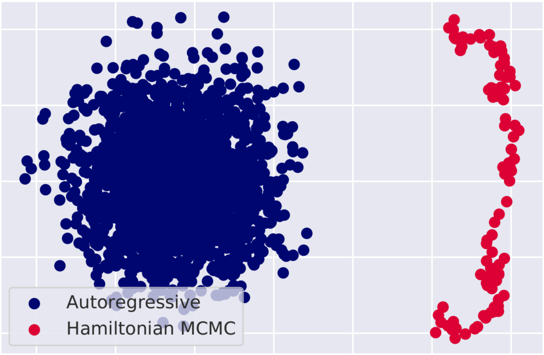

Figure 3.4: Principal component analysis of samples obtained both by auto-regressive sampling and a Hamiltonian Markov chain Monte Carlo chain for the ground-state of a 100-long transverse-field Ising chain in the disordered phase. Taken from Ref. [Vivas et al., 2022].

is the take-home message. Now, I will explain the mathematical formulas of one such auto-regressive models.

Let us start with a simple auto-regressive neural network such as the one shown in fig. 3.5. The neural network has five neurons, and you can see that four of them are used to compute the value of the fifth one. A normalisation function can be added to what results from the operation of carrying information from the first four neurons to the last one, so that the neural network can be thought of computing $q(n_5|n_1,\ldots,n_4)$, which stands for the conditional probability that n_5 takes a specific value given the first four values.

This auto-regressive model is called the chain-type Bayesian network [Frey, 1998; Wu et al., 2019, 2021; Zhao et al., 2021], which we take as the building block of the auto-regressive model. As an example, fig. 3.6 shows some auto-regressive architectures for 1D and 2D data. The input is a set of configurations, and the neural network is equipped with auto-regressive calculations. Because of the way in which neurons are connected, the output $\hat{\boldsymbol{n}}$ will contain conditional probabilities, as in fig. 3.5. In fig. 3.6, some output neuron is coloured, and its connections (also known as the receptive field of the coloured output neuron) show that the value of said neuron depends on the input neurons previous to that neuron, which is the auto-regressive property.

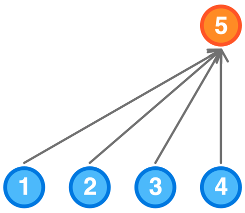

Figure 3.5: Auto-regressive network that predicts the next value in a series given the first four values. Taken from https://theblog.github.io/post/convolution-in-autoregressive-neural-networks/

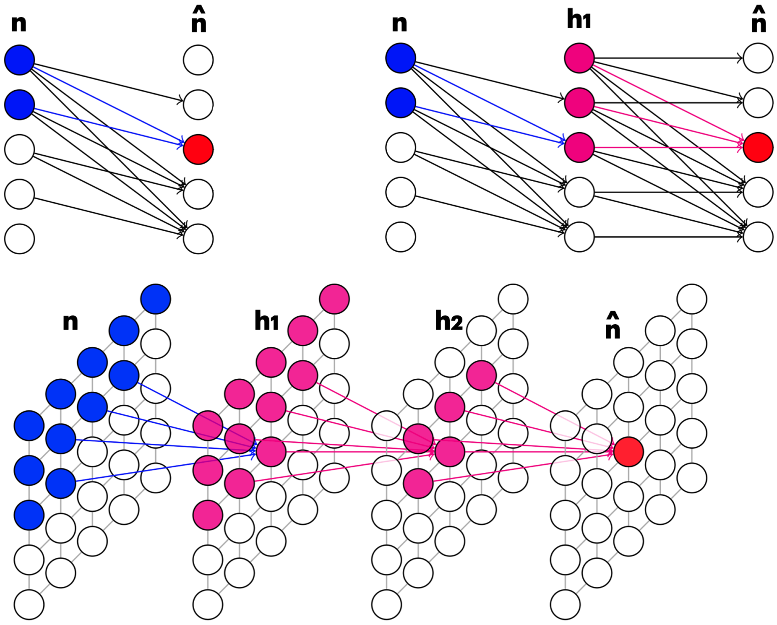

Figure 3.6: 1D and 2D architectures of auto-regressive networks. The occupation configuration $\boldsymbol{n}$ is the input to the network, $\hat{\boldsymbol{n}}$ is the output of the network which contains the conditional probabilities, and $\boldsymbol{h}$ denotes hidden layers. Adapted from Ref. [Wu et al., 2019].

With such a neural network, we can properly define the probability of a configuration as the multiplication of the output neurons in fig. 3.6:

$$P_{\boldsymbol{\theta}}(\boldsymbol{n}) = q_{\boldsymbol{\theta}}(n_1) \prod_{i=2}^{N} q_{\boldsymbol{\theta}}(n_i | n_1, \ldots, n_{i-1}). \tag{3.10}$$

Here, $q_{\boldsymbol{\theta}}$ models the probability that the i-th occupation has a specified value conditioned on the observed values for the previous occupations. These conditional probabilities are connected to the quantum mechanical state vector through the Born rule, i.e.

$$q_{\boldsymbol{\theta}}(n_i | n_1, \ldots, n_{i-1}) \equiv |\psi_{\boldsymbol{\theta}}(n_i | n_1, \ldots, n_{i-1})|^2. \tag{3.11}$$

The form of the actual wave function is an M-component conditional amplitude vector (M stands for the number of internal levels of each site; for bosons it can be any integer which truncates the maximum number of bosons per site, or for spins it is just two)

$$\boldsymbol{\eta}(n_i | n_1, \ldots, n_{i-1}) = \begin{pmatrix} \eta_1^i \\ \eta_2^i \\ \vdots \\ \eta_M^i \end{pmatrix}, \tag{3.12}$$

where $\boldsymbol{\eta}$ is an auto-regressive neural network with M complex outputs per occupation such that $\sum_{j=1}^{M} \left|\eta_j^i\right|^2 = 1$. Therefore, the actual amplitude of a configuration can be simply found by considering the vector representation of the possible values for a given occupation, which we denote as $\boldsymbol{n}_i$, which is a one-hot vector with all zeros except for the position n_i, which is one–or in the case of spins, it can be either $(1,0)^T$ for $n_i = +1$ or $(0,1)^T$ for $n_i = -1$. Thus, we write the amplitude of a configuration given by the model as

$$\psi_{\boldsymbol{\theta}}(\boldsymbol{n}) = \prod_{i=1}^{N} \boldsymbol{n}_i \cdot \boldsymbol{\eta}(n_i | n_1, \ldots, n_{i-1}). \tag{3.13}$$

3.1.3 Variational Quantum Circuits (Quantum Neural Networks)

Another important neural quantum state (which is actually a quantum neural quantum state... quantum, quantum, quantum!) is the so-called quantum neural networks, also known as variational quantum circuits [Peruzzo et al., 2014]. Variational quantum circuits are circuits with some structure of quantum gates, but the gates are parameterised by variational parameters. Essentially, the variational quantum circuit provides a parameterised unitary transformation of the state of a number of qubits (or maybe qudits in the future).

A quantum processing unit (QPU) can be composed of many individual variational quantum circuits, each of which computes the expected value of a local operator. The whole Hamiltonian is a sum of these local operators. Each of these individual variational quantum circuits are called quantum modules, and they are used to compute the expected value of said local operators. Then, the whole information is passed to a classical computer, which adds the results to get the total energy. Through the computation of gradients, the classical computer, which can be thought of a CPU, updates the parameters of the quantum circuit so as to minimise the estimated expected energy. The whole process is shown in fig. 3.7.

We did not not use variational quantum circuits, but it is important to know that they exist and how they work because I will show some results of a paper that does use them.

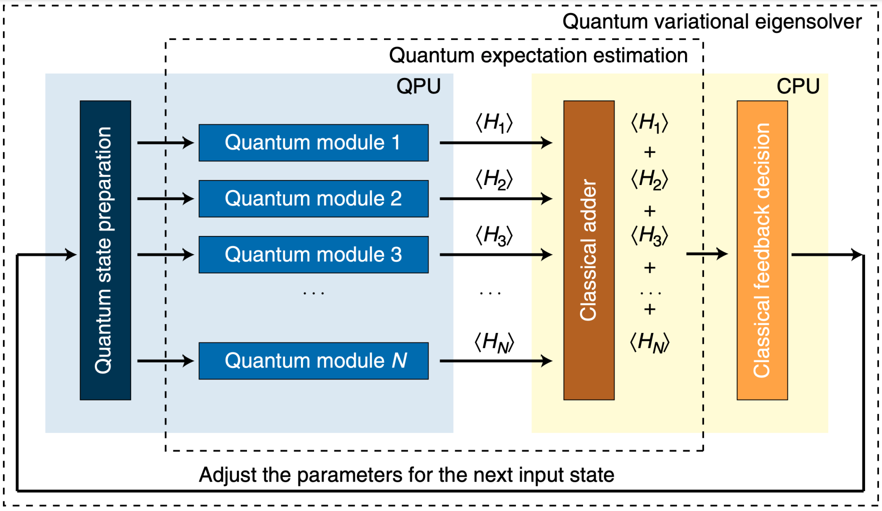
Quantum variational eigensolver
Quantum expectation estimation
QPU
CPU
Quantum state preparation
Quantum module 1
Quantum module 2
Quantum module 3
. . .
Quantum module N
$\langle H_1 \rangle$
$\langle H_2 \rangle$
$\langle H_3 \rangle$
. . .
$\langle H_N \rangle$
Classical adder
$\langle H_1 \rangle + \langle H_2 \rangle + \langle H_3 \rangle + \dots + \langle H_N \rangle$
Classical feedback decision
Adjust the parameters for the next input state

Figure 3.7: Variational quantum eigensolver. Iteratively, variational quantum circuits are used to compute expected values of local energy operators (and they can also be used to compute gradients through the so-called parameter shift rule in quantum computing [Mitarai et al., 2018]). These values are added in a CPU in order to compute the total estimated expected energy. Through an optimisation routine, the CPU sets the new parameters of the variational quantum circuits.

3.2 NETKET

The NetKet project is a fabulous open-source project for prototyping neural quantum states which has seen contributions of 43 people at the time of writing. Being open-source, it has allowed me to become a collaborator [Vicentini et al., 2021], which has taught me good scientific software development practices (for students reading this thesis, this is a super important skill to develop, it opens a lot of doors!). The idea behind NetKet is to facilitate researchers (condensed matter and also machine learning-quantum physics researchers) the study of many-body physics through neural quantum states. NetKet provides a platform for high-performance computing, and leverages variational Monte Carlo to study intractable quantum physics problems in devices such as laptops or clusters with hundreds or thousands of cores as well as graphical processing units (GPUs) due to its native use of JAX [Bradbury et al., 2018].

In this section, I will briefly explain the main components of NetKet, which allows us to perform extensive and systematic studies using neural quantum systems–I need to emphasise that, on top of NetKet, we use Optuna [Akiba et al., 2019] for hyper-parameter tuning. For a lot of intriguing details, the interested reader is encouraged to read our paper in Ref. [Vicentini et al., 2021].

3.2.1 Hilbert spaces

Hilbert spaces allow us to formally describe the state of a quantum system, from a mathematical understanding point of view. In NetKet, we can define common Hilbert spaces for several kinds of systems. For the purposes of this thesis, we only deal with discrete Hilbert spaces, for which we can have spin, qubit, Fock and fermionic orbitals Hilbert spaces. One can combine Hilbert spaces by considering their tensor product. From Hilbert spaces one can draw random configurations, which are random strings containing the occupation of each Hilbert space.

As an example, the system that we studied in chapter 2–a two-level system and an acoustic cavity–can be described by the Hilbert space of a qubit (the two-level system) and a Fock space, truncated to a number of maximum phonons.

3.2.2 Local Operators

The vast majority of Hamiltonians in quantum mechanics are given by sums of local operators because particle wave functions tend to be spatially localised, thus restricting their interaction to particles close-by.

As we saw in section 3.1.1, we need to be able to quickly compute estimated expectation values of operators, as in eq. (3.5). For this reason, NetKet supports local operator objects from which one can, given an element of the basis $\boldsymbol{b}$, obtain the non-zero matrix elements $\langle \boldsymbol{b} | A | \boldsymbol{b}' \rangle$ for every other basis element $\boldsymbol{b}'$. These are called connected elements with their respective matrix elements.

NetKet allows a seamless interface for operating operators: summing them, multiplying them by constants or by other operators, etc. Also, one can get sparse and dense representations of these operators. The way in which operators are built is an extremely efficient one. NetKet operators are ultra-sparse, meaning that they occupy even less memory than sparse matrices. The reason behind this is that the operator representation exploits the fact that many Hamiltonians are sums of local operators, which allows the sectioning of configurations into small subsystems for which connected elements can be computed very efficiently.

3.2.3 Variational Quantum States

There is a generic object, which works for closed and open systems called the variational state. It needs a set of parameters; it must have a method to compute expectation values of operators, i.e., it interfaces with local operators and neural networks; and finally, it needs a method to compute gradients of expected values, which is usually achieved through automatic differentiation (thanks to the native code in JAX, which allows immediate automatic differentiation). Optionally, for stochastic reconfiguration [Becca and Sorella, 2017] one might need to be able to compute the quantum geometric tensor (see Ref. [Vicentini et al., 2021] for more details).

The neural networks that define the variational wave function are called variational ansätze, or simply models. For numerical stability reasons, these models are built to compute the logarithm of the wave function instead of the wave function itself.

3.2.4 Samplers

In order to build the sample with which expectation values are estimated, one needs a sampler, which is an object that generates new configurations. There are two types of samplers: Markov chain Monte Carlo samplers based on the Metropolis-Hastings algorithm; or direct samplers such as an exact sampler which samples exactly from the probability distribution (you can see that this is a greedy algorithm that needs access to the whole Hilbert space), or an auto-regressive sampler, which needs an auto-regressive model.

3.2.5 Drivers

On top of everything, there is a driver which orchestrates the whole process (see fig. 3.2): it uses randomly generated configurations to estimate the energy through the expect method of the variational quantum state; then, through the gradient method of the variational quantum state, gradients of the estimated expected energy are computed in order to perform optimisation using some optimiser from machine learning or stochastic reconfiguration; after this, configurations can be proposed to build a new sample through the sampler, and the process repeats until convergence, as depicted in fig. 3.8.

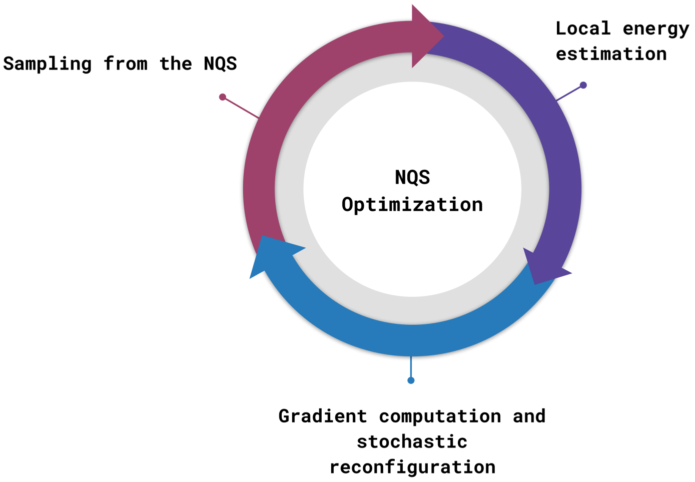

Figure 3.8: General variational Monte Carlo process. Taken from Ref. [Vivas et al., 2022].

3.3 THE GROUND STATE OF THE BOSE-HUBBARD MODEL

Nature is benevolent: even though ground states can be complicated, they usually lie on a small corner of the Hilbert space, i.e., the blot in fig. 3.2 is small. In this section, we start the exploration of the Bose-Hubbard model. This section contains excerpts and results from Ref. [Vargas-Calderón et al., 2020].

First, let us discuss the phases in the Bose-Hubbard model. The Bose-Hubbard Hamiltonian describes the interactions between bosons that can occupy sites in a d-dimensional lattice. The parameters of the model are the hopping energy t, the on-site interaction U and the chemical potential μ, so that the grand-canonical Hamiltonian reads ($\hbar = 1$) [Fisher et al., 1989]

$$H = -t \sum_{\langle ij \rangle} (a_i^\dagger a_j + \text{H.c.}) + \frac{U}{2} \sum_i n_i(n_i - 1) - \mu \sum_i n_i, \tag{3.14}$$

where a_i is the annihilation operator at site i, and $n_i = a_i^\dagger a_i$ is the number operator. The notation $\langle ij \rangle$ indicates that the sum runs over pairs of neighbour sites in the lattice, which has periodic boundary conditions.

A central issue is to calculate the quantum phase transition boundaries [Kühner and Monien, 1998; Ejima et al., 2011; Kühner et al., 2000; Batrouni and Scalettar, 1992; Kashurnikov et al., 1996; Batrouni et al., 1990; Elstner and Monien, 1999; Koller and Dupuis, 2006], whose precision has improved over the years with better calculation techniques and computing power, revealing features such as the re-entrance phenomenon [Pino et al., 2013; Elstner and Monien, 1999], where for particular values of the chemical potential, the system switches between the Mott insulator phase to the superfluid phase, and back to the Mott insulator phase before definitely entering the superfluid phase after an increase of t/U. I will explain these phases in a bit.

For simplicity, let us restrict our analyses to the $d = 1$ case, where only two quantum phases are possible (Mott insulator and superfluid). When the on-site interaction energy is much larger than the hopping energy, the latter becomes negligible, and the Hamiltonian is written as the sum of independent Hamiltonians for each site:

$$\frac{U}{2} n_i(n_i - 1) - \mu n_i. \tag{3.15}$$

These one-particle Hamiltonians can be immediately diagonalised by the number basis. The corresponding eigen-energies are $\frac{U}{2} n_i(n_i - 1) - \mu n_i$, which reach minimum values for fixed μ and U at $n_i = \max\{0, \lceil \mu/U \rceil\}$ (note that all sites are indistinguishable, and thus, equivalent). Moreover, in this regime, the expected variance of the local number operator is zero, i.e. $\langle n_i^2 \rangle - \langle n_i \rangle^2 = 0$. Physically, the bosons are restrained to each lattice site, unable to move, with a well-defined number of particles at each site. This regime characterises the Mott insulator phase. On the other hand, when the hopping energy is much larger than the on-site interaction energy, the latter becomes negligible. Thus, the Hamiltonian can also be written as the sum of independent Hamiltonians, but in momentum space, where

$$\tilde{a}_k = N^{-1/2} \sum_{j=1}^{N} a_j e^{-i x_j p_k / \hbar} \tag{3.16}$$

is the boson annihilation operator in the momentum representation. Here, $x_j = c \times j$, where c is the lattice constant, and $p_k = 2\pi k \hbar / (N \times c)$. Each independent Hamiltonian in momentum space becomes

$$\sum_k (-2t \cos(2\pi k/N) - \mu) \tilde{a}_k^\dagger \tilde{a}_k \tag{3.17}$$

and has eigen-energies $-2t\cos(2\pi k/N) - \mu$, which reach their minimum when all bosons condense with 0 momenta. Note that the energies are independent of the states' occupation, meaning that the ground state is degenerate for any number of particles. This regime is known as the superfluid phase, characterised by a delocalised wave function (formally described by algebraic decaying spatial correlations [Ejima et al., 2012; Pino et al., 2013], which is why the superfluid phase in 1D is not a true Bose-Einstein condensate). Thus, the ground state has a non-zero expected variance of the local number operator. In fact, the probability distribution for the local occupation is Poissonian, meaning that $\langle n_i^2 \rangle - \langle n_i \rangle^2 = \langle n_i \rangle$ [Greiner et al., 2003]. For a fixed chemical potential and an infinite number of sites, there exists a continuous phase transition from the Mott insulator phase to the superfluid phase as U decreases with respect to t (with the exception of a range of μ values in the first Mott lobe, where re-entrance exists, as previously described).

In Ref. [Vargas-Calderón et al., 2020] (see this work for experimental details) we performed a sweep of the Hamiltonian parameters in eq. (3.14) to reconstruct the phase diagram of the Bose Hubbard model. The systems under study were: for 5 sites, we used 8 and 20 hidden neurons in the restricted Boltzmann machine, and for 8 sites, we used 11 hidden neurons.

For the aforementioned three scenarios, the variance for the last 500 sampling-optimisation steps for variational Monte Carlo is shown in fig. 3.9(a)-(c), and the corresponding absolute errors with respect to the exact ground state energy are shown in fig. 3.9(d)-(f). It is seen that in the Hamiltonian parameter space, the majority of the energies have low-variance, showing convergence towards a value that is in excellent agreement with the exact ground-state energies.

We can compute the expected value of any observable A exactly by accessing all the probability amplitudes at every element of the canonical basis of the Hilbert space. In particular, we show in fig. 3.10 the mean absolute error of the energy with respect to the true ground state energy (notice that the difference with fig. 3.9 is that in fig. 3.9 the energy is computed with a sample, and in fig. 3.10 the energy is computed with the whole Hilbert space). It is clear that in the case of 5 sites and 20 neurons, a very large region both in the SF and MI phases shows a strong disagreement between the exact and approximated energies, showing that even though the energy converged, the state did not (correctly) converge. Another recurrent pattern is an arc of high absolute errors formed on the left-most side of the Mott lobes, which we will address in a while.

The lack of convergence of the restricted Boltzmann machine state for the case of 5 sites and 20 hidden neurons is further confirmed when we measure the overlap between the exact ground state $|\psi_{\text{exact}}\rangle$ and the restricted Boltzmann machine ground state $|\psi_{\boldsymbol{\theta}}\rangle$, shown in fig. 3.11(c). It is now clear why the expected energy with respect to the complete variational state shown in fig. 3.10(c) presents large errors when compared to the expected energy with respect to the exact ground state:

it appears that the restricted Boltzmann machine has not learned the ground state in the bottom-right region of the plot, which is a region that covers part of the Mott lobe, as well as part of the superfluid phase region. Moreover, in the case of 5 sites and 8 hidden neurons, and the case of 8 sites and 11 hidden neurons, the restricted Boltzmann machine finds difficulty in learning the ground state in the limit between the Mott insulator and the superfluid phase as shown in fig. 3.11(a) and (b).

If one pays very close attention to the energy absolute errors in fig. 3.9(d)-(f), the lowest errors occur where the overlap in fig. 3.11(a)-(c) is lowest. Therefore, even though the convergence in the energy is excellent, the state is not learned correctly–a possible occurrence in any variational method!. This is essentially, what also happened in section 3.4.4 when optimisation got stuck at excited states, even though the energy was close to the ground state energy. This difficulty in learning those states, and in general, in treating the ground state near the Mott insulator-superfluid boundary mainly arises from a Hamiltonian gap issue. In particular, in Kosterlitz-Thouless-like quantum phase transition in 1D systems [Giamarchi, 2003], an exponentially small Mott gap exists [Ejima et al., 2012], which provokes the possibility that variational methods go to the ground state energy, even when the state does not correspond to the ground state.

We see once again that there are arcs of low overlap points formed in the left-most side of the Mott lobes. Within the superfluid phase, there are also lines of low overlap, which appear because of finite size effects. Note that there are as many of these fictitious boundaries as there are sites in the periodic chain under study. Nevertheless, when comparing the 5 sites cases, it is seen that for values of $\mu/U > 1$ the states at the Mott insulator-superfluid phases boundary are better learned when 20 hidden neurons are used in the restricted Boltzmann machine (as in [McBrian et al., 2019], cf. fig. 3.11(c)) than when only 8 hidden neurons are used (see fig. 3.11(a)).

Since the advantage of variational Monte Carlo over exact diagonalization comes for intractably large Hilbert spaces, it is not always possible to compute the prob-

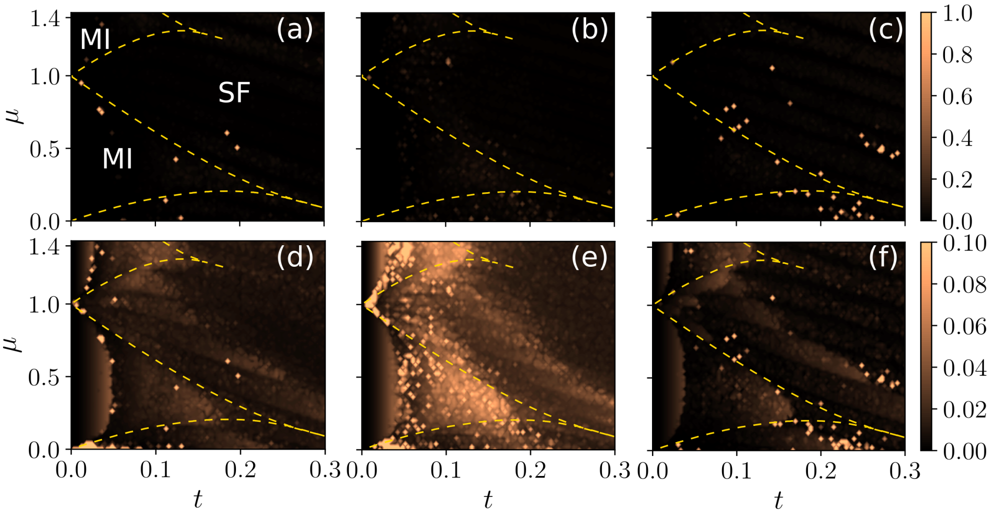

Figure 3.9: Energy variance and absolute error of the last 500 sampling-optimisation steps. Dashed lines show the phase boundaries computed for 128 sites with density matrix renormalisation group by Ejima et al. [2011]. In (a), the two first Mott lobes and the superfluid region are labelled explicitly. (a)-(c) show the variance for the last 500 sampling-optimisation steps for the cases of 5 sites and 8 hidden neurons, 8 sites and 11 hidden neurons, and 5 sites and 20 hidden neurons, respectively. (d)-(f) show the corresponding absolute errors between the average energy value for the last 500 sampling-optimisation steps and the exact ground state energy. Figure taken from Ref. [Vargas-Calderón et al., 2020].

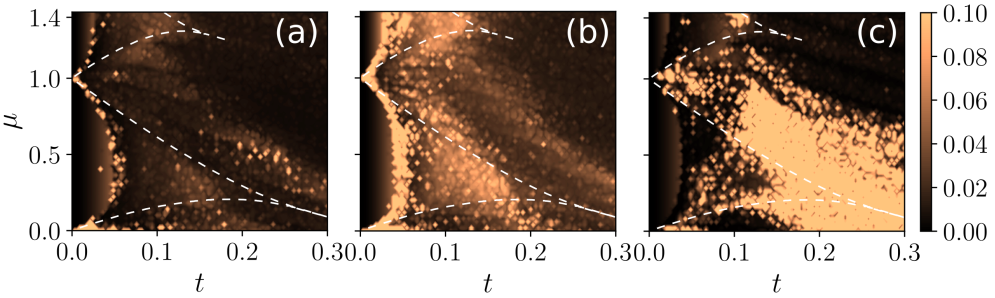

Figure 3.10: Absolute error between the RBM state expected energy and the exact ground state energy for (a) 5 sites and 8 hidden neurons, (b) 8 sites and 11 hidden neurons and (c) 5 sites and 20 hidden neurons. Dashed lines are the Mott insulator-superfluid boundaries as in fig. 3.9. Figure taken from Ref. [Vargas-Calderón et al., 2020].

ability amplitudes for all of the Fock states basis. In such a case, the restricted Boltzmann state can be cleaned as

$$|\tilde{\psi}_{\theta}\rangle = \frac{\sum_{n\in\mathcal{M}'} \psi_{\theta}(\boldsymbol{n}) |\boldsymbol{n}\rangle}{\sqrt{\sum_{n\in\mathcal{M}'} |\psi_{\theta}(\boldsymbol{n})|^2}}, \tag{3.18}$$

where $\mathcal{M}'$ is a sample of Fock states sampled with the Metropolis-Hastings algorithm, with an acceptance probability of $\min\{1, p_{\text{GC}}(\boldsymbol{n}_{i+1})/p_{\text{GC}}(\boldsymbol{n}_i)\}$, where $p_{\text{GC}}(\boldsymbol{n}) = \mathcal{Z}^{-1}\exp(-\langle \boldsymbol{n}|H|\boldsymbol{n}\rangle)$ is the probability associated with the grand canonical ensemble (note that the term $\mu\langle N\rangle$ has already been introduced in the Hamiltonian). This strategy was used to generate a sample $\mathcal{M}'$ of up to 2048 Fock states yielding a state $|\tilde{\psi}_{\theta}\rangle$ for every point in the phase diagram. The overlap between the exact ground state $|\psi_{\text{exact}}\rangle$ and the cleaned restricted Boltzmann machine ground state $|\tilde{\psi}_{\theta}\rangle$ is shown in fig. 3.12 for the three studied scenarios.

Comparing fig. 3.11 with fig. 3.12, it is seen that the retrieved sampled state "cleans" the restricted Boltzmann machine state. In fact, the Fock states whose probability amplitudes were badly learned were removed, and only the Fock states relevant for the actual ground state were left. Despite this cleaning, the larger the Hilbert space, the more states have to be sampled to consider all relevant Fock states, as it is seen that 2048 Fock states are insufficient to capture the ground state in the case of 8 sites shown in fig. 3.12(b). Nonetheless, note that the low overlap in the arc from fig. 3.11(b) almost completely disappears after the cleaning, cf. fig. 3.12(b).

On the other hand, an important feature of fig. 3.12(b) is that the overlap of the sampled restricted Boltzmann machine state diminishes as t gets larger. This happens because larger values of t/U imply larger delocalisation of the ground wave function, thus, involving more Fock states. Moreover, the Hilbert space size for 8 sites consists of $5^8 = 390625$ Fock states, which is why only sampling 2048 Fock

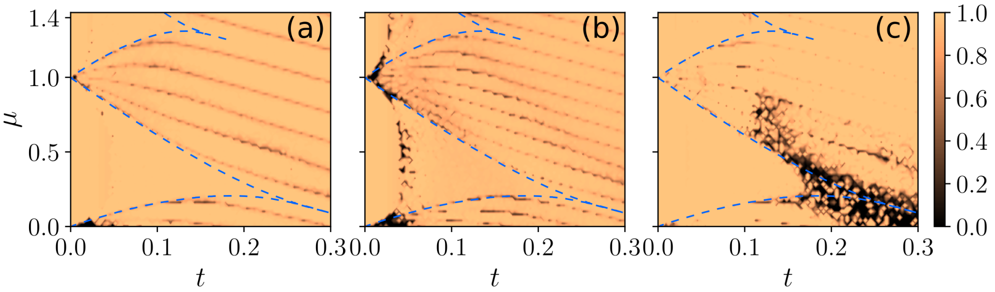

Figure 3.11: Overlap $|\langle\psi_{\text{exact}}|\psi_{\theta}\rangle|^2$ between the exact and restricted Boltzmann machine ground states for (a) 5 sites and 8 hidden neurons, (b) 8 sites and 11 neurons, and (c) 5 sites and 20 neurons. Dashed lines are the Mott insulator-superfluid boundaries as in fig. 3.9. Figure taken from Ref. [Vargas-Calderón et al., 2020].

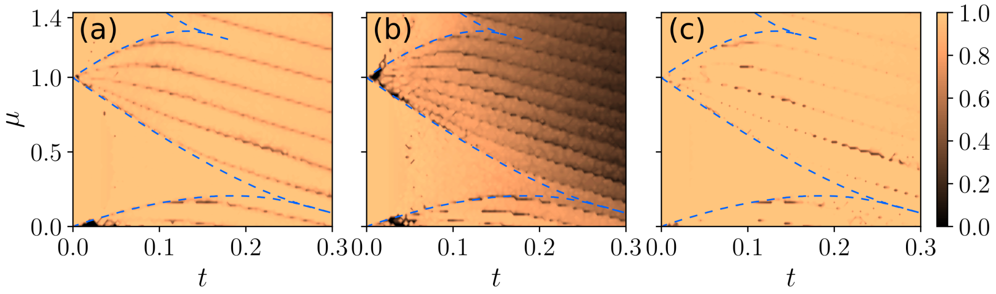

Figure 3.12: Overlap $|\langle\psi_{\text{exact}}|\tilde{\psi}_\theta\rangle|^2$ between the exact and cleaned restricted Boltzmann machine ground states for (a) 5 sites and 8 hidden neurons, (b) 8 sites and 11 hidden neurons and (c) 5 sites and 20 hidden neurons, for a maximum of 2048 states sampled from the Hilbert space. Dashed lines are the Mott insulator-superfluid boundaries as in fig. 3.9. Figure taken from Ref. [Vargas-Calderón et al., 2020].

states results in poor representations of the ground state, especially in the superfluid phase. Sampling more Fock states eventually reconstructs the exact ground state with very high overlap, except at the boundary between the Mott insulator and superfluid phases (data not shown).

Since we have shown that restricted Boltzmann machines can approximate the ground state of the Bose-Hubbard Hamiltonian in eq. (3.14), we are confident that order parameters/correlations and whatever observable comes to mind will fairly agree with the values for the exact ground state. This is obvious, since for computing any of these quantities one only needs the quantum state, which is already quite faithful. The interested reader is referred to Ref. [Vargas-Calderón et al., 2020] for further discussion on the topic, and for a complete tomography of the variational state.

3.3.1 Momentum Representation

An interesting application that we presented at the 2nd Colombian Meeting on Many-Body Quantum Simulation under the title "Machine Learning the ground state of the Bose-Hubbard model in momentum representation" in 2021 is the study of the Bose-Hubbard Hamiltonian in the momentum representation, where the Hamiltonian transforms to the form commonly encountered in second-quantisation many-body problems. The transformation in eq. (3.16) transforms the Bose-Hubbard Hamiltonian in eq. (3.14) as (we drop the tilde and just refer to momentum operators with the letters k and l)

$$H = -\sum_k (t_k + \mu) a_k^\dagger a_k + U' \sum_{k,l,k'} a_k^\dagger a_{k'}^\dagger a_l a_{k+k'-l}, \tag{3.19}$$

where there is a single momentum summation term (which actually involves all sites), and a four momentum interaction term with interaction constant $U' = U/(2N)$, whose Feynman diagram is shown in fig. 3.13.

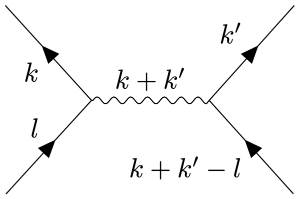

Figure 3.13: Feynman diagram for the process $a_k^\dagger a_{k'}^\dagger a_l a_{k+k'-l}$, where two particles of momenta l and $k+k'-l$ interact and are scattered with momenta k and k'.

Numerically, we saw that energy convergence is quicker (in terms of number of iterations) in the momentum representation than in the direct space representation. However, the Hamiltonian in eq. (3.19) is much more complicated (in terms of computing its connected elements and matrix elements) than its direct space version in eq. (3.14). In practice, this means that the current algorithms in NetKet run much slower. Nonetheless, I want to stress in this section that the momentum representation is relevant because Fock states in momentum space are eigenstates of the direct space translation operator. Since the system satisfies periodic boundary conditions, any translation leaves the wave function unchanged (apart from a global phase shift). Therefore, the momentum representation can naturally incorporate symmetries of the system without having to learn them or represent them explicitly in the ansatz.

Remarkably, the NetKet team has lately been keen on representation theory of symmetry groups to incorporate symmetries in ansätze, particularly with graph convolutional neural networks [Vicentini et al., 2021].

3.4 LIMITATIONS OF NEURAL QUANTUM STATES (DYNAMICAL PHASE TRANSITIONS)

Recently, Barison et al. [2022] published a fascinating work in which they used the ideas by Feynman and Kitaev to pose the problem of finding the evolution of a quantum system described by a Hamiltonian as the problem of finding the ground state of a modified Hamiltonian. Barison et al. [2022] solved the problem using variational quantum circuits. We reproduced the results, but we used neural quantum states. Let us first introduce how quantum dynamics can be approached from a ground-state perspective. This section is based on our work [Vargas-Calderón et al., 2022], and contains excerpts and results from it.

3.4.1 Feynman's Quantum Clock

Suppose that we want to evolve the quantum state of a physical system $|\psi(t_0)\rangle$ to a later time T. The formal solution of this problem is given by the Schrödinger equation, which asserts that $|\psi(T)\rangle = U(T, t_0)\,|\psi(t_0)\rangle$, where U is the time evolution operator that takes the form

$$\hat{\mathcal{T}} \exp\left(-i \int_{t_0}^{T} H(t')dt'\right), \tag{3.20}$$

with $\hat{\mathcal{T}}$ being the time ordering operator. If the Hamiltonian is time independent, it is possible to write the time evolution operator simply as

$$U(T, t_0) = U(T - t_0) = \exp(-iH(T - t_0)). \tag{3.21}$$

Let us set $t_0 = 0$. It is worth highlighting that, even though H is usually sparse, $U(T)$ need not be sparse, especially for large values of T [Alhambra, 2022]. Therefore, computing the matrix representation of $U(T)$ involves a considerable computational effort. It is much easier to compute $U(\Delta t)$ for $\Delta t = T/N \ll T$ for large N, as $U(\Delta t)$ becomes just a perturbation of the identity[4].

One of the earliest proposals for performing a quantum computation was precisely that of the time evolution of a quantum system [Feynman, 1985]. The idea behind this proposal is that the quantum state of a physical system can be described along with the quantum state of a clock [McClean et al., 2013]. In particular, the

[4] Notice that it is computationally simpler to compute $U(\Delta t)$ than $U(T)$. One can recover $U(T)$ by multiplying N consecutive times $U(\Delta t)$, but this multiplication matches the computational effort of directly computing $U(T)$

clock can be in the states $|0\rangle, |1\rangle, \ldots, |N\rangle$, i.e., it is a $N+1$-level system in the clock's Hilbert space. We encode the $N+1$-level system that describes the clock in $N_T = \lceil \log_2(N+1) \rceil$ spins. Thus, the system under consideration is enlarged with spins representing the state of the clock.

McClean et al. [2013] showed that a variational principle can help in constructing a Hamiltonian $\hat{\mathcal{H}}$ of the whole system such that its ground state is precisely

$$|\Psi\rangle = \frac{1}{\sqrt{N+1}} \sum_{t=0}^{N} |\psi(t)\rangle \otimes |t\rangle . \tag{3.22}$$

The Hamiltonian $\hat{\mathcal{H}}$ of the whole system can be written as [McClean et al., 2013; Caha et al., 2018; Barison et al., 2022]

$$\hat{\mathcal{H}} = \hat{\mathcal{H}}_0 + \frac{1}{2} \sum_{t=0}^{N-1} \left[I_{\mathcal{P}} \otimes (|t\rangle\langle t| + |t+1\rangle\langle t+1|) + \{U(\Delta t) \otimes |t+1\rangle\langle t| + \text{h.c.}\} \right], \tag{3.23}$$

where $I_{\mathcal{P}}$ is the identity of the physical system's Hilbert space $\mathcal{P}$ and $\hat{\mathcal{H}}_0 = H_0 \otimes |0\rangle\langle 0|$ is a term that breaks the degeneracy of the ground state by fixing the initial state at $t = 0$. For instance, Barison et al. [2022] take $H_0 = I - |\psi(0)\rangle\langle\psi(0)|$, for any desired initial state of the physical system $|\psi(0)\rangle$. Remarkably, the ground state of eq. (3.23) is exactly eq. (3.22), and its energy is $\langle\Psi|\hat{\mathcal{H}}|\Psi\rangle = 0$. This property is important because it allows us to quantify how close the algorithm is to converge to the true ground state of the system.

3.4.2 Transverse Field Ising Model

We studied the dynamics of the prototypical TFIM Hamiltonian defined on a one-dimensional chain of N_S (ordered) spins:

$$H = J \sum_{i=1}^{N_S-1} \sigma_i^z \sigma_{i+1}^z + h \sum_{i=1}^{N_S} \sigma_i^x , \tag{3.24}$$

where $\sigma_i^{z(x)}$ is the Z(X) Pauli operator acting only on spin i. Throughout this section, we fix the following values: $J = 0.25$ and $h = 1$.

As mentioned, the physical spin chain described by the TFIM Hamiltonian is the main system. It is enlarged by a set of spins that represent the quantum clock, as shown in fig. 3.14(a).

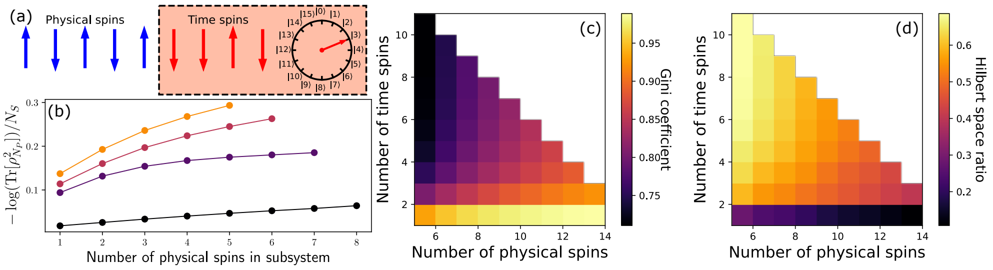

Figure 3.14: Representation of the physical spin chain enlargement with a clock state (a); and properties of the ground state of the enlarged Hamiltonian in eq. (3.23) found with exact diagonalisation. (b) shows the second Rényi entropy per spin of a sub-chain of physical spins. (c) shows the Gini coefficient of the probability distribution of the ground state, and (d) shows the ratio of the canonical basis of the Hilbert space that is needed to explain 99% of the probability of the ground state. The main text explains these plots in-depth. Figure taken from Ref. [Vargas-Calderón et al., 2022].

The ground state of eq. (3.23) encodes the time evolution of a system governed by the TFIM Hamiltonian in eq. (3.24). The initial state is set as $|\uparrow, \ldots, \uparrow\rangle$, achieved by setting $H_0 = \frac{1}{2}\sum_{i=1}^{N_S}(1-\sigma_i^z)$. There is an intrinsic difficulty in the Feynman-Kitaev Hamiltonian, which is that a lot of information (the quantum state of the Ising chain at each time step) needs to be stored in the ground state. Such difficulty is evident from analysing the structure of the ground state.

To this end, we consider a system where the total number of spins is $N_T + N_S = 9$, which fixes the Hilbert space size to 2^9 (which is perfectly tractable on a classical computer, making it easy to benchmark). We assign N_S spins to the physical TFIM, and N_T spins to encode 2^{N_T} time steps, from zero to a time $T = 3$. Let $|\Phi(N_S)\rangle$ be the ground state found through exact diagonalisation of eq. (3.23) (one actually needs to specify N_T and N_S, but we drop this because we know that their sum is 9). We can quantify the entanglement scale of the system by measuring the second Rényi entropy per physical spin $-\log\left(\mathrm{Tr}\left[\rho_{N_P}^2\right]\right)/N_S$ [Torlai et al., 2018]. Here, ρ_P is the reduced density matrix, which is obtained by tracing over all the spin degrees of freedom except the first $N_P \leq N_S$ physical spins from the ground state, i.e,

$$\rho_{N_P} = \sum_{\sigma_{N_P+1},\ldots,\sigma_{N_S+N_T}} \left\langle \sigma_{N_P+1}, \ldots, \sigma_{N_S+N_T} \middle| \Phi(N_T) \right\rangle \left\langle \Phi(N_T) \middle| \sigma_{N_P+1}, \ldots, \sigma_{N_S+N_T} \right\rangle . \tag{3.25}$$

Figure 3.14(b) shows the second Rényi entropy per physical spin, showing that the entanglement increases between the first N_P physical spins and the rest of the system as more spins are dedicated to encode time steps. Indeed, the last point of each curve in fig. 3.14(b) show that the entanglement between the time spins and the physical spins increase as long as more spins are used to encode the quantum state of the clock.

Another insightful analysis that summarises the complexity of the ground state of eq. (3.23) is the Gini coefficient of the probability distribution [Gini, 1912]. The Gini coefficient is normally used in the context of economics to assess the concentration of wealth in a society: the highest the concentration of wealth in a few individuals, the higher the Gini coefficient will be; on the other hand, it will tend to zero if wealth is perfectly distributed among the population. If we think of probability as wealth and of elements of the canonical basis as individuals, the analogy is clear. Therefore, we measure the Gini coefficient as

$$G = \frac{\sum_{\sigma,\sigma'} \left| |\langle\sigma|\Phi(N_S)\rangle|^2 - |\langle\sigma'|\Phi(N_S)\rangle|^2 \right|}{2 \times 2^9 \sum_\sigma |\langle\sigma|\Phi(N_S)\rangle|^2} . \tag{3.26}$$

We show in fig. 3.14(c) the Gini coefficient for different values of physical spins N_S and time spins N_T. The most relevant reading of this plot for our work is considering the diagonals of the plot where $N_S + N_T$ is fixed. In such a case, we see that the larger N_T is, the smaller the Gini index is. This indicates that the probability spreads across more elements of the canonical basis of the Hilbert space as N_T increases. Said fact is corroborated by fig. 3.14(d), which shows the ratio of the canonical basis elements needed to capture 99% of the probability distribution given by the ground state. More formally stated, let the canonical basis $\{\sigma\}$ be indexed such that $\left|\left\langle \sigma^{(i)} \middle| \Phi(N_S) \right\rangle\right|^2 \geq \left|\left\langle \sigma^{(i+1)} \middle| \Phi(N_S) \right\rangle\right|^2$. Then, fig. 3.14(d) shows the ratio $r/2^{N_S+N_T}$, where r is the smallest integer such that $\sum_{i=1}^{r} \left|\left\langle \sigma^{(i)} \middle| \Phi(N_S) \right\rangle\right|^2 < 0.99$. It is clear from fig. 3.14(d) that for a fixed number of total spins $N_S + N_T$, the larger N_T is, the highest the ratio of elements in the canonical basis needed to explain the ground state is.

Figure 3.14 showed that the ground state, for large values of N_T, is not anymore in a "small corner" of the Hilbert space. Instead, a well-spread and highly entangled ground state forms.

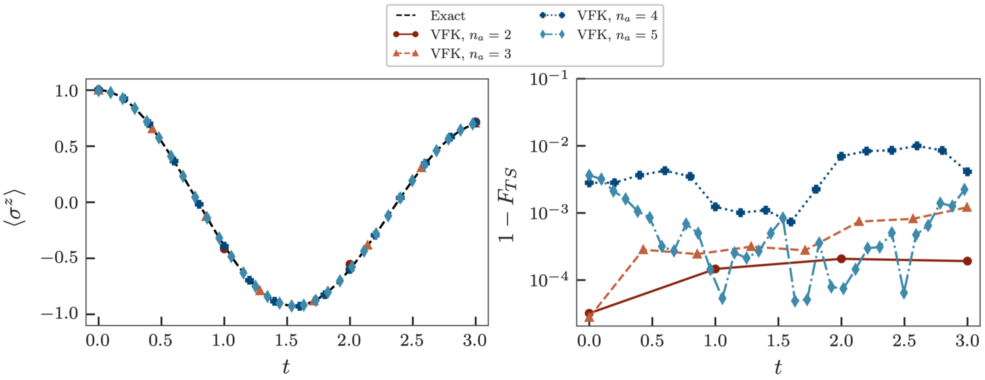

Figure 3.15: Average magnetisation and infidelity of the "variational Feynman Kitaev" (VFK) ansatz (the name that Barison et al. [2022] give to their variational quantum circuit) for a different number of auxiliary qubits n_a, which induce 2^{n_a} time steps in the evolution. Figure adapted from Ref. [Barison et al., 2022].

3.4.3 Ground States and Dynamical Phase Transition by Quenching

We will now see how Barison et al. [2022] use variational quantum circuits to learn said ground state, and, in the next subsection, how we use neural quantum states to do the same. In essence, we see that both variational quantum circuits and neural quantum states are able to find the ground states. This representation ability is referred to as expressivity. However, our large numerical experiments show that neural quantum states are much more difficult to train, and that learning ground states becomes a limitation of another aspect of machine learning models: trainability. We see that the models can be expressive enough, but training them is a different story.

The reader is referred to Ref. [Barison et al., 2022] for the exact variational quantum circuit used by them to simulate the ground state of eq. (3.23). Once this ground state $|\Psi_{\boldsymbol{\vartheta}}\rangle$ is obtained, where, in this case, $\boldsymbol{\vartheta}$ are the variational parameters of the quantum circuit, one can obtain the state of the main spin chain by simply projecting onto a particular clock time:

$$|\psi(t)\rangle = \sqrt{N+1}\,\langle t|\Psi\rangle\,, \tag{3.27}$$

which easily follows from eq. (3.22). From having the quantum state at different points in time, we can measure any observable. In fig. 3.15, the average magnetisation and infidelity with respect to exact states[5] are shown. You can see that infidelities are pretty low and that the magnetisation curve matches perfectly with the exact one.

With this setup, dynamical phase transitions [Heyl, 2018; Heyl et al., 2013] can be studied through the formalism of Loschmidt echoes. The Loschmidt echo is defined as

$$L(t) = |\langle\psi(0)|\psi(t)\rangle|^2, \tag{3.28}$$

which allows us to identify dynamical quantum phase transitions through the rate of change of the Loschmidt echo with respect to the system size:

$$\lambda(t) = -\lim_{N_S\to\infty}\frac{1}{N_S}\log L(t). \tag{3.29}$$

In particular, dynamical quantum phase transitions can be identified by the non-analytic behaviour of $\lambda(t)$ at some particular time.

[5] Exact actually refers to a simulation under the Trotter-Suzuki decomposition of time evolution, due to reasons explained in Ref. [Barison et al., 2022]; in this thesis we assume that evolution under the Trotter-Suzuki decomposition is exact, because it approximates the exact solution with great precision.

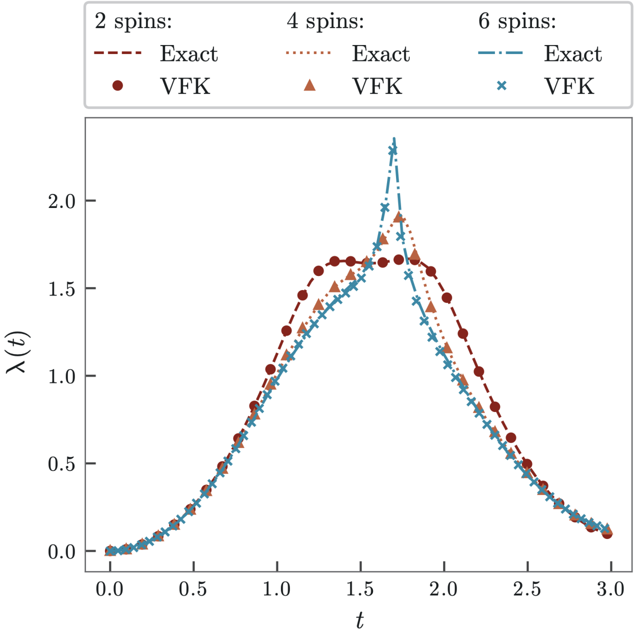

Figure 3.16: Rate function of the Loschmidt echo for the TFIM in eq. (3.24) measured using the variational quantum circuit state with the Feynman Kitaev ansatz. The plot shows a dynamical phase transition at t near $\pi/2$, which comes as a non-analytic behaviour of the curves as the number of spins increases. Figure taken from Ref. [Barison et al., 2022].

Barison et al. [2022] prepare the initial state to be the trivial all-spins-up state, and, at time zero, they turn on the external magnetic field in eq. (3.24), letting the system evolve, just as shown in fig. 3.15. They observe the scar of a dynamical phase transition as they increase the number of spins in the spin chain of the TFIM, as shown in fig. 3.16.

Training the variational circuits, at least for these system sizes, is quick and accurate. However, we emphasise that these results were obtained through the simulation of the variational quantum circuit on a classical computer, which enables the exact access to the variational state $|\psi_\vartheta\rangle$. Therefore, the variational energy $E^\star_\vartheta$ is not estimated, as it would be on a real quantum device, but can be computed exactly. The same occurs with the gradients of the variational energy with respect to variational parameters. On a real quantum device, however, these quantities have to be estimated, which is both time-consuming and introduces inaccuracy. Moreover, scalability of the variational quantum eigensolver setup to study large spin chains might be endangered by trainability issues in variational quantum circuits [Anschuetz and Kiani, 2022] such as the onset of barren plateaus [McClean et al., 2018].

3.4.4 Trainability Issues of Neural Quantum States

When trying to reproduce the results by Barison et al. [2022], but using neural quantum states, we found a lot of difficulty. We decided to make extensive and systematic training of neural quantum states using restricted Boltzmann machines and auto-regressive models. These extensive study was performed with Optuna [Akiba et al., 2019] for hyper-parameter tuning.

Both variational Monte Carlo and the neural quantum state training have hyper-parameters that dictate the behaviour of the variational energy optimisation. Hyper-parameter tuning is a difficult meta-optimisation task that, in our case, thrives to answer the questions: what is the best structure of the training algorithm, and what is the neural quantum state architecture that produces the lowest variational energy?

The two main components of variational Monte Carlo (as presented in NetKet, see section 3.2) are the sampler and the optimiser. The sampler dictates how the sample $\mathcal{M}$ of eq. (3.5) is built, and the optimiser is a rule for updating the parameters $\boldsymbol{\theta}$ of the neural quantum state.

As mentioned in some parts of this chapter, it is common to find that stochastic reconfiguration (SR) [Sorella et al., 2007] is used in combination with stochastic gradient descent (SGD) as an optimiser. SR takes into account the geometry of the variational energy landscape to update parameters in the directions that yield maximum descent. However, we experimented on optimisation instances that used the restricted Boltzmann machine (eq. (3.9)) with different numbers of hidden neurons using both SR+SGD and AdamW [Loshchilov and Hutter, 2017] and found no significant difference in performance. On the contrary, AdamW was faster, which is why we chose it as the optimisation method for all of the experiments shown in the main text. We consider its learning rate as the only hyper-parameter of the optimiser. Regarding the sampler, we consider the number of parallel Markov chains and the number of total samples as its two hyper-parameters. In the case of an auto-regressive neural quantum state, no Markov chains are considered, and the sampler only has the number of total samples hyper-parameter.

The hyper-parameters for the architecture of the neural quantum states are different for the restricted Boltzmann machine and the auto-regressive ansätze. For the restricted Boltzmann machine, the hyper-parameter is $\alpha := N_H/(N_S + N_T)$, which specifies the proportion of hidden neurons with respect to the visible neurons. For the auto-regressive ansatz, the auto-regressive neural network $\boldsymbol{\eta}$ in eq. (3.12) has two hyper-parameters: the number of layers N_L, and the number of hidden neurons N_H of each layer, with the property that the layers are masked in such a way that the conditional probability of a spin taking a value depends only on the values of the previous spins. The hyper-parameter tuning algorithm that we used is the tree-structured Parzen estimator (TSPE) [Bergstra et al., 2011] provided in the Optuna package [Akiba et al., 2019]. The priors for each hyper-parameter are shown in table 3.1. Hyper-parameter tuning was conducted for 100 different hyper-parameter sets for each ansatz, and for each combination of number of physical spins N_S and number of time spins N_T.

Figure 3.17 shows the increasing difficulty of training a restricted Boltzmann machine for the ground state of eq. (3.23) through variational Monte Carlo as the number of time spins N_T grows. We fixed the total number of spins $N_S + N_T = 9$, as in fig. 3.14(b). For each value of N_T (between 1 and 4), we performed hyper-parameter tuning for 100 iterations with Optuna [Akiba et al., 2019], aiming to minimise the variational energy in eq. (3.5). Figure 3.17(a)-(d) show the time evolution of the spin chain for the hyper-parameters that produced the smallest infidelities.

Table 3.1: Prior distribution of hyper-parameters for the tree-structured Parzen estimator for the restricted Boltzmann machine (RBM) and auto-regressive (AR) models. Table taken from Ref. [Vargas-Calderón et al., 2022].

NQS	Hyper-parameter	Support	Prior
RBM & AR	Number of samples	$[256, 2048]$	Uniform
	Learning rate	$[10^{-4}, 1]$	Loguniform
RBM	Number of Markov chains	$\{4, 8, 16\}$	Uniform
	α	$\{1, \ldots, 5\}$	Uniform
AR	N_L	$\{1, 2, 3\}$	Uniform
	N_H	$\{2, 4, 8, 16, 32\}$	Uniform

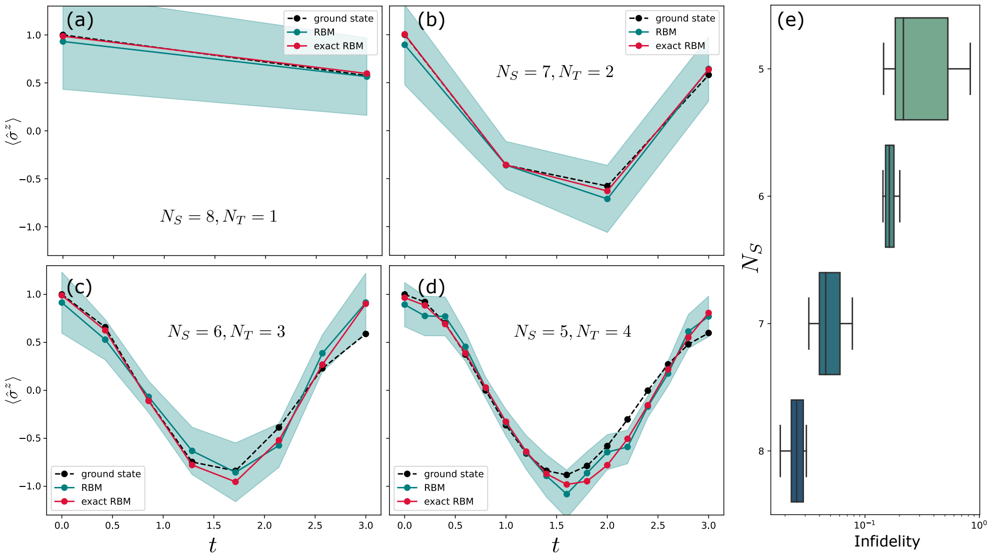

Figure 3.17: Time evolution approximated with a restricted Boltzmann machine. (a)-(d) show the expected value of the average magnetisation $\langle \sigma^z \rangle = \frac{1}{N_S} \sum_{i=1}^{N_S} \langle \sigma_i^z \rangle$. In each panel curves are shown for the average magnetisation obtained through exact diagonalisation (ground state), estimation of the variational magnetisation with a sample $\mathcal{M}$ (RBM) and exact variational magnetisation (exact RBM), which results from using the complete state vector instead of a sample. The shaded region indicates the estimated fluctuations of magnetisation using the sample $\mathcal{M}$. The lines serve as guide for the eye only. (e) shows a box plot of infidelity $(1 - |\langle \Phi(N_S) | \psi_{\boldsymbol{\theta}^\star} \rangle|^2)$ for the best 10 hyper-parameter experiments, where $|\Phi(N_S)\rangle$ is the exact ground state and $\boldsymbol{\theta}^\star$ indicates that parameters have been optimised until convergence. Figure taken from [Vargas-Calderón et al., 2022].

It is seen in fig. 3.17(a)-(d) that, overall, the evolution of the average magnetisation is in accordance to the average magnetisation obtained through exact diagonalisation for all the values of N_T. In the plots, the exact RBM line refers to magnetisation measured using the complete state vector from the restricted Boltzmann machine, instead of estimating the magnetisation through a sample. However, the qualitative agreement of magnetisation curves does not exhibit the difficulty of training the restricted Boltzmann machine as N_T increases. Indeed, the infidelities for fig. 3.17(a)-(d) are 0.018, 0.032, 0.144 and 0.145, respectively. The increasing infidelity, as N_T grows, indicates that training becomes more difficult, despite the Hilbert space always having the same size. However, these are only the best states found after hyper-parameter tuning. Figure 3.17(e) shows a box plot of the infidelities of the best 10 hyper-parameter tuning states, where a clear trend appears: the larger N_T is, the more difficult it is to find the correct ground state.

Regarding the auto-regressive ansatz in eq. (3.13), we see in fig. 3.18 qualitatively similar, but worse performance in terms of correctly describing the evolution of magnetisation. This is further confirmed by poor infidelities when N_T is large. The lowest infidelities achieved after hyper-parameter tuning were 0.025, 0.063, 0.517 and 0.850 for fig. 3.18(a)-(d), respectively. Even though one can sample directly from the probability distribution induced by the auto-regressive ansatz, avoiding issues with the Markov chain sampling, it is clear that capturing the ground state of eq. (3.23) is more challenging.

Compared to the variational quantum eigensolver method, variational Monte Carlo struggles with finding an accurate approximation of the true ground state, presenting infidelities at least one order of magnitude higher than infidelities reported by Barison et al. [2022]. Unlike variational Monte Carlo, the variational quantum eigensolver approach by Barison et al. [2022] directly handles a normalised quantum state in the whole Hilbert space, and its parameterisation consists of lo-

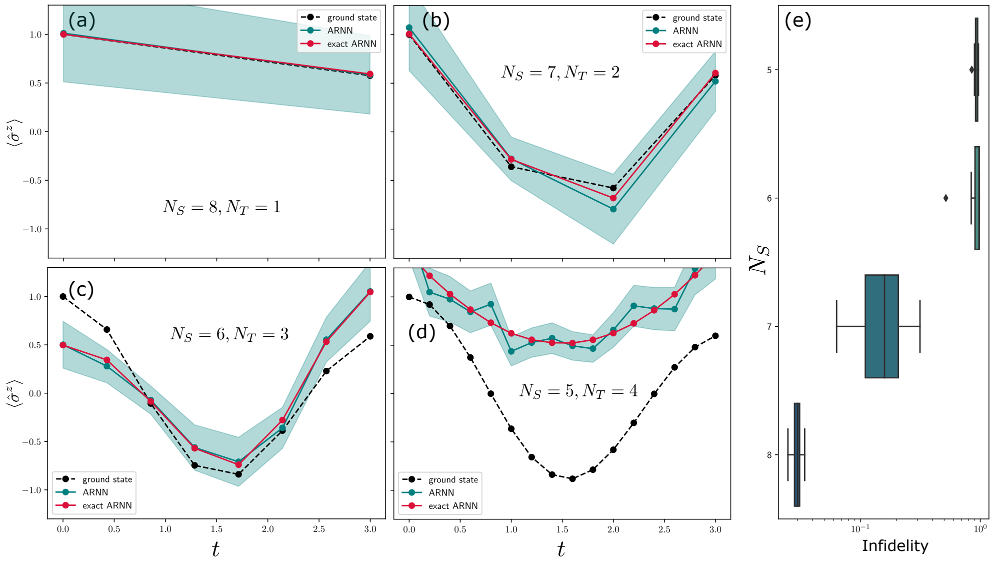

Figure 3.18: Similar to fig. 3.17 but for the auto-regressive ansatz (ARNN) in eq. (3.13). Figure taken from Ref. [Vargas-Calderón et al., 2022].

cal transformations that preserve the norm. A natural set of questions that arise are: what is it that makes neural quantum states have larger infidelities than the variational quantum eigensolver? Is it expressivity? Is it trainability? [Wright and McMahon, 2020; Abbas et al., 2021] If a neural quantum state can represent the ground state of eq. (3.23) with low infidelity, it means that the neural quantum state is expressive enough, but trainability hampers the possibility of describing the correct ground state, as shown in fig. 3.17.

Considering the previous discussion, let us explore the expressivity of the restricted Boltzmann machine. The most challenging experiment tackled in this section is the one of $N_S = 5$ and $N_T = 4$, which is perfectly tractable for a classical computer. We consider the problem of finding parameters for the restricted Boltzmann machine ansatz that are able to faithfully describe the ground state of eq. (3.23). To this end, we directly minimise the infidelity $1 - |\langle \Phi(N_S = 5) | \psi_{\boldsymbol{\theta}} \rangle|^2$, where the restricted Boltzmann machine state $|\psi_{\boldsymbol{\theta}}\rangle$ is directly accessed, i.e., no sampling is involved. Experimentation with the ansatz in eq. (3.9) shows that training leads to local minima of the infidelity landscape, hinting convergence to stable excited states of eq. (3.23). Therefore, we turned over to a similar restricted Boltzmann machine ansatz, which defines one restricted Boltzmann machine for the modulus of the wave function and another for the phase of the wave function, namely the Modulus-Phase-RBM or MP-RBM [Torlai et al., 2018]

$$\psi_{\boldsymbol{\theta}}^{\text{MP-RBM}} = \exp\left(\psi_{\boldsymbol{\theta}_{\text{Re}}} + i\psi_{\boldsymbol{\theta}_{\text{Im}}}\right), \tag{3.30}$$

where $\psi_{\boldsymbol{\theta}_{\text{Re}}}$ and $\psi_{\boldsymbol{\theta}_{\text{Im}}}$ are restricted Boltzmann machines defined by eq. (3.9), with real-only parameters $\boldsymbol{\theta}_{\text{Re}}$ and $\boldsymbol{\theta}_{\text{Im}}$.

Training the MP-RBM ansatz in eq. (3.30) to minimise the estimated variational energy (see eq. (3.5)) with variational Monte Carlo yields similar infidelities than the restricted Boltzmann machine ansatz after hyper-parameter tuning (0.160 for the $N_S = 5, N_T = 4$ case). However, it is easier to train the MP-RBM when minimising the infidelity (even without hyper-parameter tuning). In fact, we see that the MP-RBM is capable of learning the ground state with an infidelity of 2×10^{-3}, as depicted by the excellent agreement between the MP-RBM magnetisation curve and the exact one in fig. 3.19.

Figure 3.19 exhibits the probability distribution of each state induced onto the canonical basis of the Hilbert space. Since the infidelity-optimised MP-RBM leads to a very low infidelity, differences between its distribution (top middle panel) and

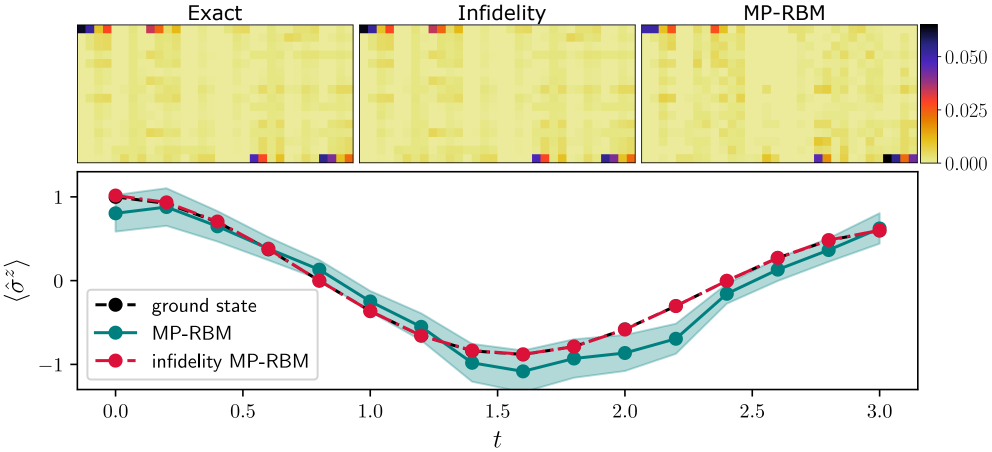

Figure 3.19: Probability of each element of the canonical basis of the Hilbert space and time evolution of magnetisation for an MP-RBM ansatz. The top panel shows the 2^9 probabilities associated to each element of the canonical basis for the ground states obtained through exact diagonalisation, through variational minimisation of the infidelity, and through variational minimisation of the estimated energy in the left, middle and right sub-panels, respectively. The bottom panel shows the average magnetisation obtained with each of these states, where the "ground state" line corresponds to the magnetisation obtained with exact diagonalisation, the "MP-RBM" line is obtained through variational minimisation of the estimated energy, and the "infidelity MP-RBM" is obtained through variational minimisation of the infidelity. Figure taken from Ref. [Vargas-Calderón et al., 2022].

the exact ground state distribution (top left panel) are minimal. However, differences with the MP-RBM obtained through VMC are larger. This is reflected onto the average magnetisation curves, shown in the bottom panel of fig. 3.19. These results are in agreement with the study by Deng et al. [2017], who show that neural quantum states based on restricted Boltzmann machines have a wide expressivity, able to represent many highly entangled quantum states.

Similar results are obtained for the auto-regressive model. Again, we find that optimising the infidelity for the ansatz in eq. (3.13) traps the neural quantum state into an excited state of eq. (3.23), which is why we turned over to an auto-regressive ansatz that explicitly divides the modulus and phase of the wave function, similar to the works by Hibat-Allah et al. [2020] and Barrett et al. [2022]. In this setup, we divide the auto-regressive neural network η into two auto-regressive neural networks, one for the modulus, and the other for the phase of the wave function. Training this ansatz to minimise the estimated variational energy with variational Monte Carlo, yields high infidelity of 0.920 after hyper-parameter tuning for the $N_S = 5, N_T = 4$ case, which is a similar infidelity to the one obtained by the ansatz in eq. (3.13). On the other hand, the direct optimisation of infidelity (without any hyper-parameter tuning) yields an infidelity of 3.5×10^{-3}, comparable to that of the MP-RBM. This result further supports the fact that neural quantum states are able to accurately approximate highly entangled ground states with widely-spread probability distributions. The problem, however, resides on trainability when performing VMC. An important final remark is that knowing the reason why infidelity optimisation consistently fails for neural quantum states with complex-only parameters remains an open question.

3.5 THE STEADY-STATE OF THE DRIVEN-DISSIPATIVE BOSE-HUBBARD MODEL

Common experimental setups that show a Bose-Hubbard-type Hamiltonian, are ions in optical traps or photons in arrays of Kerr resonators. In this section we need a clear experimental setup in mind to be able to relate to all the different processes that I am going to be talking about.

Each site of the Bose-Hubbard model in a lattice can be one of such Kerr resonator cavities, which corresponds to a physical system that holds a bosonic mode. The Hamiltonian of such a system (a single cavity) is

$$H_0 = \omega_c n + \frac{U}{2} n(n+1), \tag{3.31}$$

where ω_c is the characteristic energy of the cavity, U is the boson-boson interaction inside the cavity, and n is the number operator. We further consider a pumping mechanism whereby bosons are injected coherently into the cavity, with an associated Hamiltonian

$$\hat{H}_P = F(e^{i\omega_p t}\hat{a}^\dagger + e^{i\omega_p t}\hat{a}), \tag{3.32}$$

where F is the amplitude of the pump[6]. We restrict ourselves to a one-dimensional linear lattice of N cavities, so we index each bosonic operator by i to refer to each cavity. Since bosons can hop between neighbouring positions of the lattice, we consider a characteristic hopping energy t, so that the complete Hamiltonian, in the rotating frame of the pump, is written as

$$H = \sum_{i=1}^{N} \left(-\Delta n_i + \frac{U}{2} n_i(n_i+1) + F(a_i^\dagger + a_i) \right) + \frac{t}{2} \sum_{i=1}^{N-1} \left(a_i^\dagger a_{i+1} + a_{i+1}^\dagger a_i \right). \tag{3.33}$$

In eq. (3.33), the parameter Δ is a detuning between the pump frequency ω_p and the cavity frequency, i.e., $\Delta = \omega_p - \omega_c$. Furthermore, we consider Markovian dissipative channels with one-particle decay collapse operators. Thus, the density operator is described by the GKSL equation [Lindblad, 1976; Gorini et al., 1976]

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = \mathcal{L}\rho = -i[H,\rho] + \gamma \sum_{i \in D} \left(a_i \rho a_i^\dagger - \frac{1}{2}\left\{ a_i^\dagger a_i, \rho \right\} \right), \tag{3.34}$$

where $\mathcal{L}$ is the Liouvillian super-operator. $D \subseteq \{1, \ldots, N\}$ is a set that selects the cavities onto which dissipation acts with a rate γ. Experimentally, this can be achieved by means of controlling the coupling of each cavity to the environment [Barontini et al., 2013; Dogra et al., 2019]. In this section we consider odd N, and we focus on global dissipation ($D_\mathrm{G} = \{1, \ldots, N\}$) and local dissipation ($D_\mathrm{L} = \{(N+1)/2\}$).

Before jumping in to see some results about this exploration with neural quantum states, let us take a look at fig. 3.20. This is the phase diagram of the mean field version of eq. (3.34), when considering global dissipation. The mean field approximation transforms the Hamiltonian to

$$H_\mathrm{MF} = -\Delta a^\dagger a + \frac{U}{2} a^\dagger a^\dagger a a + (F - t\phi)a^\dagger + (F - t\phi^*)a, \tag{3.35}$$

where $\phi = \langle a \rangle$ is the so-called superfluid parameter.

Figure 3.20 shows that there are structures similar to the Mott lobes, which do not become as well defined as in the closed case. The figure, however, points us to a nice region where we can study the problem beyond mean field approximations.

[6] For simplicity, we take the pump amplitude F to be real.

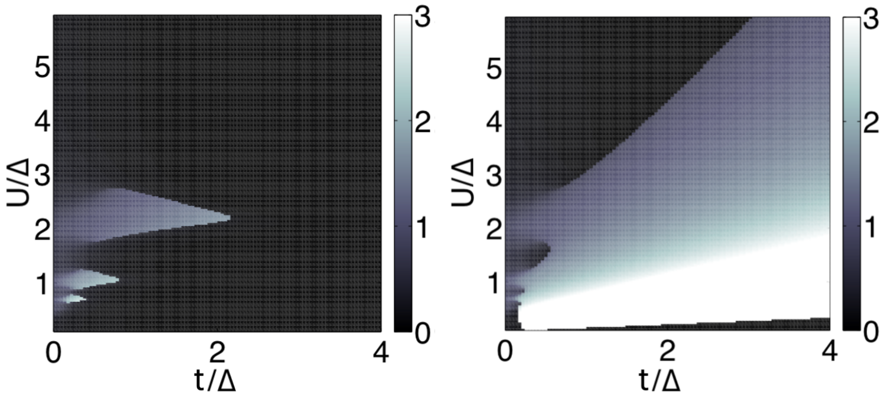

Figure 3.20: Phase diagram of the mean field driven-dissipative Bose-Hubbard showing the expected number of photons for $F/\Delta = 0.4$ and $\gamma/\Delta = 0.2$. The number of photons has been truncated to a maximum of three, for clarity. Left panel shows the so-called low-density phase, and right panel shows the high-density phase, which are the two solutions for the bi stability that comes from the mean field equations. Figure adapted from [Le Boité et al., 2013].

In particular, we chose $U = 1.5\Delta$, and we vary the tunnelling rate t, where for small t we can go from a Mott insulator-like phase to a superfluid-like phase in the high-density phase or in between Mott-like lobes.

Regarding neural quantum states, it is clear that in the context of open quantum systems, the Hamiltonian can no longer be studied independently from the dissipative channels. Instead, the central object that describes the quantum system is the Liouvillian shown in eq. (3.34). Similarly, the ground state is no longer well-defined because the state of the system is not a state vector anymore, but a density matrix. In this sense, the ground state delegates its central role to the non-equilibrium steady state (NESS) of the system, which is crucial to investigate several properties of the quantum mechanical system [Breuer et al., 2002], as well as phenomena such as open quantum phase transitions [Minganti et al., 2018].

The Liouvillian is a completely positive and trace-preserving map, but is, in general, non-unitary [Albert and Jiang, 2014; Manzano and Hurtado, 2018]. The theoretical study of the Liouvillian has established that its complex spectrum is bound to the left quadrants of the complex plane, indicating that the real part of the eigenvalues is always negative or 0 [Spohn, 1976, 1977]. In particular, there is always an eigenvalue which is exactly 0, and its corresponding eigenmatrix is the NESS of the physical system [Baumgartner and Narnhofer, 2008].

The seminal papers that introduced neural quantum states for NESS search [Nagy and Savona, 2019; Vicentini et al., 2019; Yoshioka and Hamazaki, 2019] exploit the spectral structure of the Liouvillian to define the optimisation problem to solve. A neural quantum state that models the NESS of a quantum mechanical system described by the Liouvillian $\mathcal{L}$ is $\rho_{\text{NESS}} \approx \min_{\theta} \text{Tr}\left[\rho_{\theta}^{\dagger}\mathcal{L}^{\dagger}\mathcal{L}\rho_{\theta}\right]$.

Thus, we can describe how to use variational Monte Carlo to estimate the NESS of an open quantum system by making an analogy with the closed quantum system case: $\mathcal{L}^{\dagger}\mathcal{L}$ takes the place of the Hamiltonian $\hat{A}$ and the neural quantum state is now a density matrix ρ_{θ}. More explicitly, we can estimate $\text{Tr}\left[\rho_{\theta}^{\dagger}\mathcal{L}^{\dagger}\mathcal{L}\rho_{\theta}\right]$ as a cost function [Vicentini et al., 2019]

$$C(\boldsymbol{\theta}) = \left\langle \sum_{\boldsymbol{n}',\tilde{\boldsymbol{n}}'} \left| \mathcal{L}(\boldsymbol{n},\tilde{\boldsymbol{n}};\boldsymbol{n}',\tilde{\boldsymbol{n}}') \frac{\rho_{\boldsymbol{\theta}}(\boldsymbol{n}',\tilde{\boldsymbol{n}}')}{\rho_{\boldsymbol{\theta}}(\boldsymbol{n},\tilde{\boldsymbol{n}})} \right|^2 \right\rangle_{(\boldsymbol{n},\tilde{\boldsymbol{n}})\in\mathcal{M}} . \tag{3.36}$$

In eq. (3.36) we keep the indices $\boldsymbol{n}$ to denote elements of the canonical basis of the Hilbert space $\mathcal{H}$, but we promote these indices to $(\boldsymbol{n},\tilde{\boldsymbol{n}})$, which denote elements of the doubled Hilbert space $\mathcal{H} = \mathcal{H} \otimes \mathcal{H}$. Therefore, we refer to the matrix elements of ρ_{θ} as $\rho_{\theta}(\boldsymbol{n},\tilde{\boldsymbol{n}})$. Similarly, the Liouvillian $\mathcal{L} : \mathcal{B}(\mathcal{H}) \to \mathcal{B}(\mathcal{H})$ is a super-operator which acts on the space of density matrices $\mathcal{B}(\mathcal{H})$, and its elements are referred to as $\mathcal{L}(\boldsymbol{n},\tilde{\boldsymbol{n}};\boldsymbol{n}',\tilde{\boldsymbol{n}}')$. Finally, as in the closed quantum system case, we build a sample $\mathcal{M} \in \mathcal{H}$ that can be obtained, for instance, through MCMC, and is distributed according to $|\rho_{\boldsymbol{\theta}}(\boldsymbol{n},\tilde{\boldsymbol{n}})|^2$.

Now, let us focus on the neural quantum state that will be used to approximate the NESS. We consider the ansatz proposed by Torlai and Melko [2018], which is based on the idea that a (mixed) density matrix can always be obtained by tracing out degrees of freedom of a system. In spite of being applicable for spin Hilbert spaces, we leverage this ansatz to higher-dimensional Hilbert spaces such as the ones needed to describe boson occupations in the Bose-Hubbard model.

Our Hilbert space can be written as $\mathcal{H} = \bigotimes_{i=1}^{N} \mathcal{F}$, where $\mathcal{F}$ is the Fock space of a single cavity. Computationally, we truncate the Fock space to allow for a maximum number of bosons $n_{\max}$. Similarly as in Ref. [Vargas-Calderón et al., 2020], as we explained in section 3.3 we encode the occupation of each cavity to a set of spins. In this section, we use binary encoding, which uses $\lceil \log_2 n_{\max} \rceil$ spins to describe the occupation of each cavity. Then, our transformed Hilbert space is $\mathcal{H}_B = \bigotimes_{i=1}^{N_S} \mathcal{S}$, where $\mathcal{S} = \{\alpha_1 |\uparrow\rangle + \alpha_2 |\downarrow\rangle : |\alpha_1|^2 + |\alpha_2|^2 = 1\}$ is the Hilbert space of a spin and $N_S = N\lceil \log_2 n_{\max} \rceil$ is the total number of physical spins.

Now, we apply the ideas of Torlai and Melko [2018] by enlarging the Hilbert space of the physical system with the Hilbert space $\mathcal{H}_A = \bigotimes_{i=1}^{n_a} \mathcal{S}$ of an auxiliary set of n_a spins. The state of the enlarged system is described by a pure neural quantum state given by a (tripartite) restricted Boltzmann machine (RBM) [Torlai et al., 2018; Torlai and Melko, 2018], so that $\rho_{\boldsymbol{\theta}}^{\mathcal{H}_B \otimes \mathcal{H}_A} = |\psi_{\boldsymbol{\theta}}\rangle\langle\psi_{\boldsymbol{\theta}}|$, where $|\psi_{\boldsymbol{\theta}}\rangle$ takes the explicit form of

$$\sum_{\boldsymbol{\sigma},\boldsymbol{a}} \psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma},\boldsymbol{a})\, |\boldsymbol{\sigma}\rangle \otimes |\boldsymbol{a}\rangle\,. \tag{3.37}$$

Here, $\boldsymbol{\sigma}$ is the spin configuration corresponding to the binarisation $\boldsymbol{n} \mapsto \boldsymbol{\sigma}$ of the elements of the canonical basis of $\mathcal{H}$; and $\boldsymbol{a}$ denotes the elements of the canonical basis of $\mathcal{H}_A$.

The mixed state of the physical system can be found by tracing out the auxiliary degrees of freedom, which can be done analytically due to the structure of the tripartite RBM [Torlai and Melko, 2018; Vicentini et al., 2019]:

$$\rho_{\boldsymbol{\theta}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}) = \exp\Big(\Gamma^{+}_{\boldsymbol{\lambda}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}) + i\Gamma^{-}_{\boldsymbol{\mu}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}) + \Pi_{\boldsymbol{\theta}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})\Big). \tag{3.38}$$

The parameters $\boldsymbol{\theta}$ consist of $\{W^{\eta}_{j,\ell}, U^{\eta}_{j,\varsigma} b^{\eta}_{j}, c^{\eta}_{j}, d_{\varsigma}\}$, for $\eta = \lambda, \mu$. The index j runs from 1 to the number of physical spins N_S, and the indices ℓ and ς run from 1 to the number of hidden spins $N_H^B =: \alpha N_S$ and $N_H^A =: \beta N_S$, respectively, where we also defined the proportions of hidden spins α and β [Vicentini et al., 2019]. The neural networks in eq. (3.38) are given by [Torlai and Melko, 2018; Vicentini et al., 2019]

$$\begin{aligned}\Gamma^{\pm}_{\boldsymbol{\eta}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}) = \frac{1}{2}\Bigg[\sum_{j=1}^{N_S} b^{\eta}_{j}(\sigma_j \pm \tilde{\sigma}_j) + \sum_{\ell=1}^{N_H^B} \log\cosh\left(c^{\eta}_{\ell} + \sum_{j=1}^{N_S} W^{\eta}_{j,\ell}\sigma_j\right) \\ \pm \sum_{\ell=1}^{N_H^B} \log\cosh\left(c^{\eta}_{\ell} + \sum_{j=1}^{N_S} W^{\eta}_{j,\ell}\tilde{\sigma}_j\right)\Bigg],\end{aligned} \tag{3.39}$$

$$\Pi_{\boldsymbol{\theta}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}) = \sum_{\varsigma=1}^{N_H^A} \log\cosh\left(d_{\varsigma} + \frac{1}{2}\sum_{j=1}^{N_S} U^{\lambda}_{j,\varsigma}(\sigma_j + \tilde{\sigma}_j) + \frac{i}{2}\sum_{j=1}^{N_S} U^{\mu}_{j,\varsigma}(\sigma_j - \tilde{\sigma}_j)\right). \tag{3.40}$$

We can study eq. (3.34) with exact methods, but only with 3 cavities and a maximum of 4 photons per cavity, otherwise our 1/4 terabyte RAM nodes run out of memory and calculations become incredibly slow. The case of global dissipation is shown in fig. 3.21. In comparison with fig. 3.20, it looks like the system (beyond the

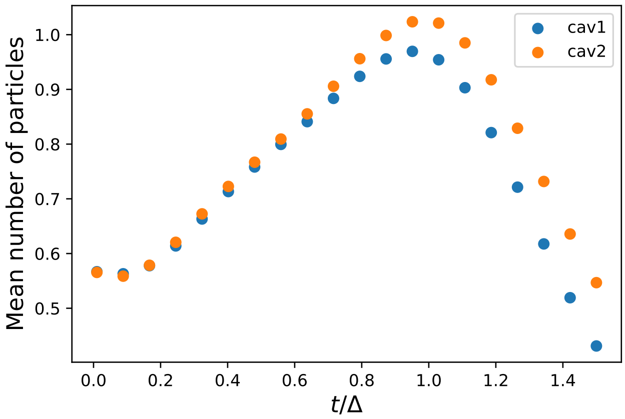

Figure 3.21: Mean occupation for first and second cavity in a line of three cavities for varying tunnelling rate in the steady state obtained through exact diagonalisation.

mean field approximation) resembles a low-density phase, where the occupation grows as the tunnelling increases, but then it goes down again, as the tunnelling keeps increasing, depleting the system from particles. Note that only cavity 1 and cavity 2 occupations are shown because cavity 1 and 3 are completely equivalent, but 2 is not, as it has two neighbours instead of just one. Also, in all cases, the occupation of cavity 2 is lower than that of cavity 1.

The case of local dissipation is much more complicated. Using an exact solver yields disparate occupations. The reasons for numerical instability are unknown. Therefore, we did not find the steady state through exact diagonalisation, but we did so through the rather normal strategy of evolving the state until it no longer changes, namely, we impose the condition that the Hilbert distance between 10 different states in the last 1000 time steps must not be larger than a threshold set to 10^{-6}. fig. 3.22 shows the evolution of the mean number of particles for small and large values of t. It can be seen that a large value of t stabilises the evolution much faster than a small value of t.

By considering the same sampled points of t in fig. 3.21, we plot the mean number of particles in the steady state for local dissipation in fig. 3.23. For low values of t, there are some small inaccuracies, but overall the side cavities have more particles than the cavity in the middle. This is to be expected, as for low tunnelling, the side cavities receive photons from the pumping laser, but they cannot transfer those to the central cavity, which is the only particle sink in the system. However, after a given tunnelling rate, photons start to flood the middle cavity from the sides and from the laser that pumps photons into it: the photon escape mechanism can no longer keep up, and the mean number of photons in the middle cavity become larger than in the side cavities, as is also the case for global dissipation.

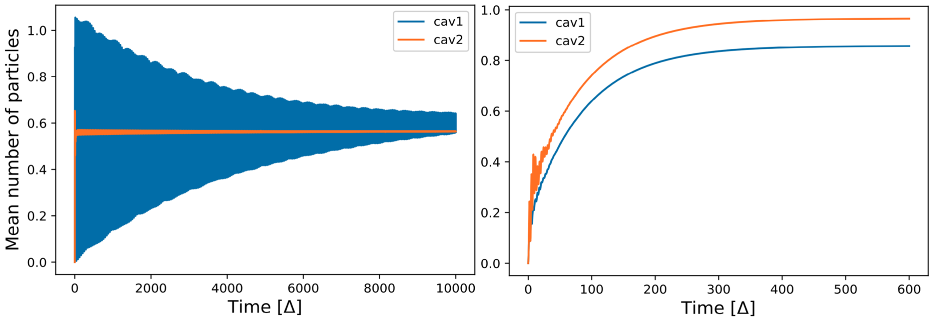

Figure 3.22: Evolution of the mean number of particles for $t/\Delta = 0.01$ (left) and $t/\Delta = 1.5$ (right).

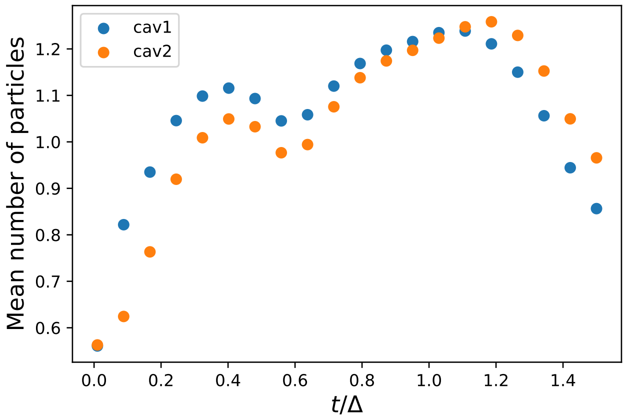

Figure 3.23: Mean occupation for first and second cavity in a line of three cavities for varying tunnelling rate in the steady state obtained through evolution of the trivial $|0,0,0\rangle$ state.

We performed extensive numerical experiments in the low tunnelling regime ($t = 0.01\Delta$) and high tunnelling regime ($t = 1.5\Delta$) using the neural quantum state eq. (3.38). For each of these cases, hyper-parameter tuning with Optuna [Akiba et al., 2019] was carried out. Before delving into the results, I need to express that this problem is way more difficult than the closed case, and choosing a right optimiser can make the difference (even though in the end it did not make the difference in terms of the neural quantum state quality).

Since the introduction of neural quantum states [Carleo and Troyer, 2017], stochastic reconfiguration (SR, also known as natural gradient in the machine learning community) has been the *de facto* approach to train these variational ansätze. Despite its conceptual clarity, SR is costly with respect other baseline optimisation algorithms, which is why optimisation methods that perform well, such as AdamW [Loshchilov and Hutter, 2017], are very popular in the machine learning community. On the other hand, the physics community is mostly interested in using algorithms that find lower minima. In principle, if SR offers such an advantage, we should choose it as the optimisation strategy over other options. However, as we show next, we see no evidence that SR offers lower minima with respect to AdamW. In fact, AdamW achieves similar performance as SR, but is generally faster.

The experiment that we performed for our particular problem at hand is the following. We generated 100 random driven-dissipative Bose-Hubbard systems characterised by the following parameters. The number of cavities N was chosen between 3 and 8. The onsite interaction U, the hopping interaction t, the pump amplitude F and the dissipative rate γ were chosen uniformly between $[0.1, 4]$, $[0.01, 2]$, $[0.1, 1]$ and $[0.01, 0.4]$, respectively. Parameters regarding variational Monte Carlo with the neural quantum state as a variational ansatz were chosen as follows. The size of the neural quantum state was determined by α and β (the ratio of visible and auxilliary neurons in the deep Boltzmann machine defined by eq. (3.38)), which were selected from the set $\{1, 2, 3\}$, with the restriction that $\alpha + \beta \leq 4$. The learning rate and standard deviation of the initial parameters were generated from a log-uniform distribution on $[10^{-4}, 0.5]$ and $[10^{-5}, 1]$, respectively. Other parameters were fixed: the detuning $\Delta = 1$, the diagonal shift for stochastic reconfiguration was 0.01, the number of samples and diagonal samples for the variational Monte Carlo iterations were $200N$ and $75N$, respectively. Once a set of parameters was defined, the optimisation was performed 5 times for different random seeds. Figure 3.24 shows the performance and duration differences between experiments using SR and AdamW. It is clear that AdamW is, in general, considerably faster than SR. Despite being more experiments for which SR outperforms AdamW, the difference is negligible

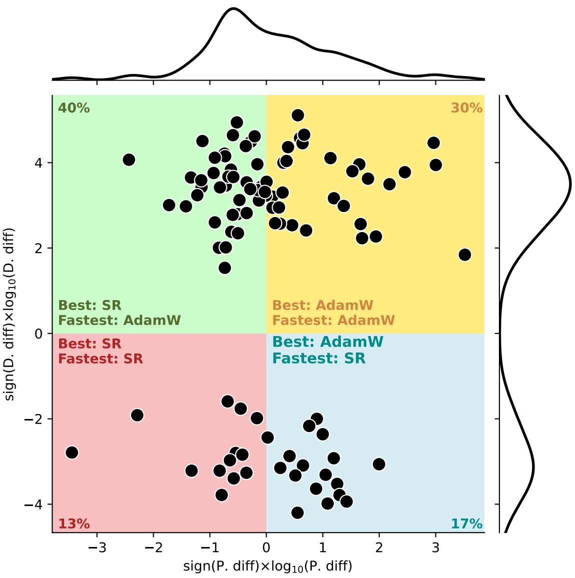

Figure 3.24: Performance difference versus duration difference between stochastic gradient descent with SR and AdamW. The performance difference is defined as $\overline{C(\boldsymbol{\theta}_{\text{SR}})} - \overline{C(\boldsymbol{\theta}_{\text{AdamW}})}$ (see eq. (3.36)). The duration difference is defined as the time took by SR experiments minus the time took by AdamW experiments, in seconds. Kernel density estimation plots are shown for both performance and duration differences. Percentages of experiments on each quadrant are also shown.

(note that fig. 3.24 is in logarithmic scale). Thus, we use AdamW throughout this section.

Now, as mentioned earlier, we performed hyper-parameter tuning to find the steady state of the system in eq. (3.33) for a number of cavities. We can visualise two-dimensional cuts of the explored hyper-parameter space, to give a taste of how these experiments are carried out. In particular, since training is a stochastic process, we collected a small, but telling sample of 5 different training curves for each hyper-parameter set. We show in fig. 3.25 this visualisation example for the simpler case of three cavities. It can be seen that only the learning rate defines a clear region where results are better. The rest of the parameters interact in complicated manners.

Table 3.2 presents the mean occupations of the left-most cavity and the middle cavity in one-dimensional chains of cavities described under the driven-dissipative Bose Hubbard model in eq. (3.33) for a low value of tunnelling. When comparing

Table 3.2: Mean occupation of the side left-most cavity and the middle cavity for different chain sizes with local and global dissipation from the best neural quantum state after hyper-parameter tuning, when $t = 0.01\Delta$. The confidence interval is in parenthesis.

	Number of cavities		
Local Dissipation	3	5	7
Side cavity	3.0078(78)	3.039(17)	2.097(25)
Middle cavity	2.254(205)	0.117(51)	0.047(33)
Global Dissipation	3	5	7
Side Cavity	0.238(50)	1.46(16)	0.99(12)
Middle cavity	0.227(41)	1.45(16)	0.85(13)

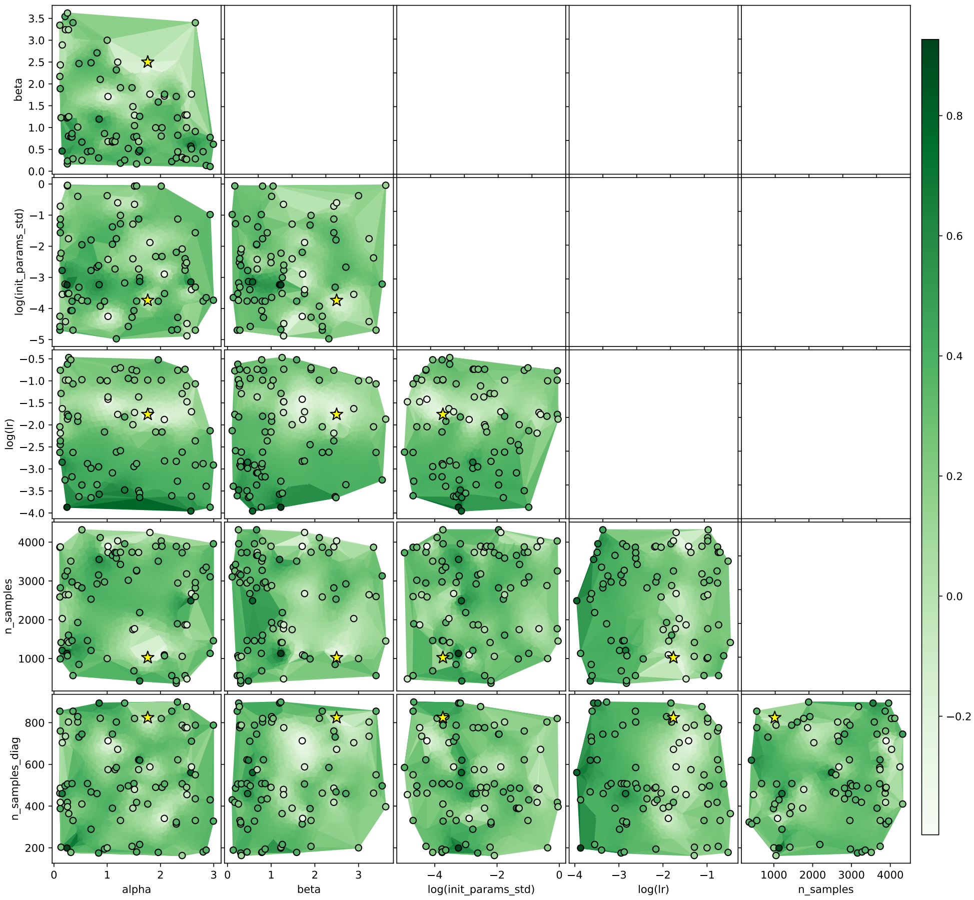

Figure 3.25: Two-dimensional cuts of the hyper-parameter space for three cavities. The scale shows the logarithm in base 10 of the average cost in eq. (3.36) in 5 realisations of training with different initial random seeds. The yellow star indicates the best set of hyper-parameters. The hyper-parameters under consideration are (from top to bottom, and from left to right without repetitions) the ratio of auxilliary units in the deep Boltzmann machine, the logarithm of the standard deviation of the initial parameters of the deep Boltzmann machine, the learning rate, the number of samples and number of diagonal samples at each variational Monte Carlo iteration, and the ratio of hidden units in the deep Boltzmann machine.

these values for 3 cavities with figs. 3.21 and 3.23 it is clear that they are not compatible, signalling the lack of quality of the neural quantum state. The fact that for local dissipation the occupation of the side cavity is very near an integer might

Table 3.3: Mean occupation of the side left-most cavity and the middle cavity for two chain sizes with local and global dissipation from the best neural quantum state after hyper-parameter tuning, when $t = 1.5\Delta$. The confidence interval is in parenthesis.

	Number of cavities	
Local Dissipation	3	5
Side cavity	0.48(11)	2.30(17)
Middle cavity	0.57(11)	1.80(13)
Global Dissipation	3	5
Side Cavity	0.96(15)	2.76(16)
Middle cavity	1.07(15)	2.15(13)

suggest the collapse of the neural quantum state to a particular sector of the Hilbert space, which traps the probability distribution and does not let it explore the complete space. Furthermore, no trends arise apart from the decreasing mean number of photons in the middle cavity when local dissipation is present.

In the case of high tunnelling (see table 3.3), the situation does not change much: neither local nor global dissipation are well approximated by the neural quantum state as can be seen solely from the expected number of photons in the cavities.

These drawbacks from the neural quantum states for open quantum systems are far worse than the errors found in the spin systems that were considered in the seminal papers of Yoshioka and Hamazaki [2019]; Vicentini et al. [2019]; Nagy and Savona [2019]. It might be the case that open bosonic problems are too hard for this neural quantum state, or it might also be the case that minimising eq. (3.36) is not the best approach to the problem. In any case, this exploration of ours is an important piece of evidence that points at a limitation of the ansatz in eq. (3.38). Even though other neural quantum states can be used, we believe that another approach is desirable. Why? From the theory of open quantum phase transitions [Minganti et al., 2018] we know that the Liouvillian gap closes at critical points (see fig. 3.26), which will make minimising eq. (3.38) exponentially difficult (with respect to the inverse size of the gap). Therefore, other approaches–such as the one we took with the exact study of the three cavity case for local dissipation–are needed. In particular, we believe that time evolution is a promising alternative. However, approaches such as the time-dependent variational Monte Carlo [Carleo et al., 2017; Hartmann and Carleo, 2019], which is already experimentally available in NetKet[7], does not perform well for states that are highly peaked (i.e. their probability is highly concentrated on one or very few elements of the canonical basis of the Hilbert space). The reason for this is that equations do not allow a clean transport of the probability.

What other options do we have? This is actually the motivation for our work in section 3.4. We want to perform short time evolution in order to leverage the quantum trajectories technique (see section 2.3). To do this, we lay out two long-term projects for future research:

- Construct a Computer Algebra System that facilitates performing symbolic quantum mechanical operations in a computer. As of now, NetKet deals with local operators, but the Feynman-Kitaev approach requires the use of projectors. A projector is an operator far away from a local operator, which is why NetKet's interface renders insufficient. With a computer algebra system we would be able to deal with local operators as in NetKet, but we could also easily handle projectors in the canonical basis. The development of this project is already being carried out by myself at Zapata Computing Inc, but escapes the scope of this doctoral work.

- The ability to use neural quantum states to find ground states of non-hermitian operators. As you recall from section 2.3, one needs to evolve a system with a non-hermitian operator. Whatever ansatz is used for this purpose, one must be able to also collapse the wave function with a measurement (given by the collapse operator), which means that clever updates of parameters need to be done to reflect the action of collapse operators on the neural quantum state *a la* Jonsson et al. [2018].

3.6 SUMMARY AND OUTLOOK

This chapter has shown how machine learning can help solving difficult problems in quantum mechanics. The great expressivity and variability of ansätze based on neural networks has been so far used to provide accurate approximations of

[7] https://netket.readthedocs.io/en/latest/api/api-experimental.html

ground states. However, we see that trainability remains an issue for very difficult ground states where the probability is spread through the entire Hilbert space, or in cases such as open quantum systems, where the number of configurations is truly intractable.

Nonetheless, we point out the fact that neural quantum states become important alternatives to quantum computers in the task of studying quantum mechanics computationally. Even with the presence of fault-tolerant quantum computers, algorithms for estimating energies (or Liouvillian real spectra) still require an unfeasible number of measurements. We urge the community of neural quantum states to formally and experimentally investigate the scaling of sample complexity and error. Also more diverse models with new takes on handling intractable Hilbert spaces are needed to overcome the current difficulties of studying ground states and steady states in closed and open quantum systems, respectively. This thesis contributes to the advancement of understanding the field and reflecting on the capabilities of neural quantum states. We are optimistic that the collaboration of tens, hopefully hundreds of physicists around the world will continue advancing neural quantum states further as a tool to study quantum phase transitions in the usual set up of a closed system, dynamical quantum phase transitions, and open quantum phase transitions. This progress will also likely advance understanding in machine learning. As mentioned before, Herbert's group is already working towards these goals with another PhD student and a master's student. This research line will probably continue to grow, as more physics students become interested in data science and thus their knowledge of machine learning is ever increasing.

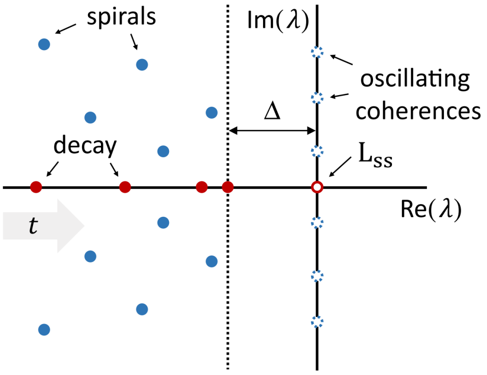

Figure 3.26: Eigenvalues of the Liouvillian. All eigenvalues satisfy $\mathrm{Re}(\lambda_i) \leq 0$ [Evans, 1976]. Blue points are eigenvalues whose corresponding eigenmatrix will have an oscillating, yet decaying behaviour as time evolves. Red points are eigenvalues whose corresponding eigenmatrix will only damp as time evolves. White points with blue edges are steady-state coherences, which oscillate with frequency $\mathrm{Im}(\lambda_i)$. Finally, white points with red edges are steady-states, and represent the eigenmatrices that span the null-space of the Liouvillian. The Liouvillian gap Δ measures a characteristic decay time that separates any initial state of the system from the final steady-state. The larger the gap is, the faster the system's state converges to the steady-state. Figure taken from [Albert and Jiang, 2014].

4 QUANTUM-INSPIRED MACHINE LEARNING

In this chapter we explore the other direction of the machine learning-quantum physics avenue. In chapter 3 we used machine learning for quantum physics, now we use quantum physics for machine learning. This chapter presents a framework for performing machine learning tasks using the mathematical tools normally used in quantum mechanics. The idea underneath is simple: quantum mechanics' central object is the wave function, which describes the state of a physical system by taking into account its (quantum) probabilistic nature. In this part of the thesis, we propose such a framework that uses the way in which quantum mechanics expresses probability distributions to model classical probability distributions, which are at the core of machine learning and the statistical sciences. Importantly, our framework defines composable and differentiable blocks, which will come in handy for research in future years in many areas of the data sciences because we can combine our approach with deep learning models.

After the presentation of the state of the art, in section 4.2 I explain the rudiments of our framework for supervised learning problems. This approach was extended to unsupervised learning problems in section 4.3, and finally, I show in section 4.4 how we implemented all of our framework in real quantum circuits using IBM's quantum computers named "Lima" and "Bogotá".

4.1 STATE OF THE ART

The works that we will present in this chapter approach the task of learning from classical data using mathematics from quantum mechanics. Mainly, we focus on algorithms that are optimised to run on classical computers. But, as we will show, our framework can be also run on quantum devices. Nonetheless, we make the distinction that we will always deal with classical data and our framework is quantum mechanical regardless of it running on a classical or a quantum computer. In a broad sense, our work belongs to the area of quantum machine learning, which is the combination of data generated by either quantum or classical experiments and algorithms which are quantum in the sense that they are either inspired by quantum mechanics or they naturally run on quantum devices. Some authors, such as Schuld and Petruccione [2018], prefer to restrict quantum machine learning as algorithms that purely run on quantum computers (or are devised to be run on quantum computers), but we consider that any algorithm strongly based on quantum mechanics but runs on a classical computer is also a quantum machine learning algorithm. We can see a depiction of the division between different approaches to quantum machine learning in fig. 4.1.

The combination of classical methods and quantum methods for machine learning has been almost exclusively focused on the application of classical optimisation strategies to parameterised operations in quantum hardware [Cerezo et al., 2021]. However, there is a largely unexplored area: classical machine learning methods which are quantum-inspired, which our work addresses. This area is mainly bounded by the interpretation of probability theory in light of quantum mechanics [Fine, 1973; Wilce, 2021]. One of the principal discussions addressed by Wilce [2021] is that projections in quantum mechanics are operations that allow the computation of probabilities, thus providing a way to perform probability calculations with projection operators. This will be the departure point and a central theme in the framework that we develop in this chapter.

I need to mention that during discussions and research with Fabio, other groups were also looking at the specific problem of classification with a framework similar to ours. Such is the work of Sergioli et al. [2021]; Tang [2019]; Tiwari and Melucci [2019]; Giuntini et al. [2021], which is essentially based on the idea that data of different classes can be mapped to quantum states such that quantum states of different classes are distinguishable. Classification is thus performed through quantum state discrimination.

Our approach is novel within the field of quantum machine learning, particularly because, as we will see, its formulation allows a completely optimisation-free approach to learning tasks. This is of paramount importance, because there are many examples of both quantum machine learning and regular machine learning algorithms, where learning is heavily hampered by theoretical reasons. Let me explain this. The most popular approach to learning is gradient-based optimisation of a cost function that one wishes to minimise. Under usual circumstances, one has a parameterised model, which is a function with parameters that can be tuned to solve a problem which involves a cost function. What is normally done is that the cost function depends on values of said parameterised model. It follows from calculus that the minimum of the cost function can be reached when the gradients of the cost function with respect to the parameters of the model are zero.

However, models can involve thousands, millions, or more parameters, with the additional difficulty that models can be highly non-linear. Therefore, finding a set of parameters for which the gradients of the cost function are zero is practically impossible because of the intractability of the problem. Nonetheless, the machine learning community has developed, for decades, impressive optimisation algorithms that are able to reach very good parameters that minimise (at least locally) the cost function. The algorithms that update the parameters depend on the magnitude (and also variance) of the gradients. It is the case that in many scenarios in both regular machine learning and quantum machine learning, gradients become prohibitively small, rendering optimisation algorithms completely useless. This is particularly pervasive in quantum machine learning through the phenomenon of barren plateaus [McClean et al., 2018], whereby gradients become exponentially small with respect to the size of the quantum circuit at hand. Our work, on the other hand, has the possibility of not involving gradient-based optimisation at all, and as we will show, this results in perfectly fine results. Nonetheless, we easily extend the framework to having

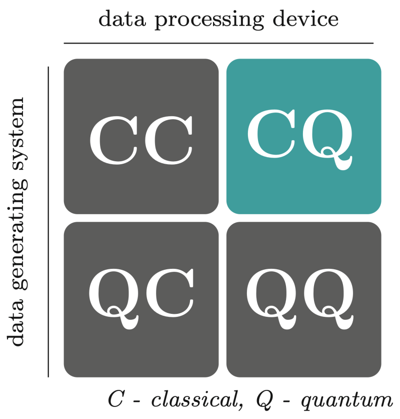

Figure 4.1: Quantum machine learning subdivision. The chart shows combinations of classical/quantum data and classical/quantum device where algorithms are run. Our work belongs to the top row, where we deal with classical data and we run algorithms on either classical or quantum computers, but the algorithms are fully quantum mechanical-inspired. Taken from Ref. [Schuld and Petruccione, 2018].

gradient-based optimisation components which can improve performance in many applications, as we also show.

Before presenting the framework, which explores the fertile and largely unexplored field of quantum-inspired machine learning, we present the usual tasks that are solved in machine learning.

4.1.1 The Problem of Learning

Commonly, machine learning problems are divided in supervised and unsupervised problems. Supervised problems are those in which one wants to predict a property of an object or an entity, and is given a data set of objects for which such property is known. The job in supervised learning is to find patterns in the data set that allows a high-quality prediction of the property under consideration for some object from which we know its properties but the property under consideration. For example, one can be given a data set containing the following features

- Waist perimeter (cm)
- Chest perimeter(cm)
- Thighs perimeter (cm)
- Height (cm)
- Weight (kg)
- Age (years)
- Biological sex

for a population of human beings. Now, one can ask, what is the biological sex of a new human being for which all the properties but sex are previously known? This problem is known as a (binary) classification problem, where an instance is required to be classified into one of two classes: male and female.

Also, one can ask, what is the age of a new human being for which all the properties but age are previously known? This problem is known as a regression problem, where an instance is required to be assigned a number in a continuous interval. These sort of problems where one asks a value for a feature which is known for a population are supervised problems because we have a reference value for which we can measure the errors in our prediction. These data sets are known as training data sets, because they allow to train a model or an algorithm that learns how the feature in question is related to the other features, and how can the other features be used and combined to yield an accurate prediction of the feature under consideration.

In many natural sciences, the most common supervised learning problem is the so-called curve fitting problem (one can argue that most problems are curve fitting problems), where one has a data set $\{(x_i, y_i)\}_{i=1}^N$ of N data points and one wants to find a function f such that $f(x_i) = y_i$. Of course, there are infinitely many such functions, which is why methods such as cross-validation are useful to measure the generalisation capability of a model f. Generalisation is a very complicated philosophical problem, but we have some tools to deal with it. Essentially, we can split our data set into a purely training data set and a testing data set: consider the disjoint sets I and J so that $I \cup J = \{1, \ldots, N\}$, then the training data set is $\{(x_i, y_i)\}_{i \in I}$ and the testing data set is $\{(x_j, y_j)\}_{j \in J}$. One trains a model f using the information of the training data set so that for $i \in I$ one has $f(x_i) \approx y_i$ (in reality we thrive for $\approx\ \to\ =$) such that $f(x_j) \approx y_j$ for $j \in J$. This means that we want to keep the error of predicting the training data set small while at the same time having a model f that "generalises" to unseen (by the model) data, which are in the testing data set.

One normally defines a parameterised model $f_{\boldsymbol{\theta}}(x)$, where $\boldsymbol{\theta}$ is a vector of parameters of the model. An easy example is a linear regression model. Say that we promote the variable x to a vector in $\mathbb{R}^d$, so that in our data set $x_i \to \boldsymbol{x}_i$. A linear regression model is $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_d x_d + \theta_{d+1}$, where we have abused the notation a bit, and x_k here are the components of the vector $\boldsymbol{x}$. A logistic regression model, for instance, applies a non-linearity to the linear regression model, wrapping it with a sigmoid function. You can see that any model restricts the behaviour of the function: the functional form of the model defines a subset of functions in the function space of all functions from $\mathbb{R}^d$ to $\mathbb{R}$. Going back to no vectors, i.e., restoring the training and testing data sets, we can measure the errors in the training and testing data sets as

$$E_{\text{train}} = \sum_{i \in I} L(x_i, y_i), \qquad\qquad E_{\text{test}} = \sum_{j \in J} L(x_j, y_j), \tag{4.1}$$

with, for example, $L(x,y) = \|f_{\boldsymbol{\theta}}(x) - y\|^2$. As mentioned, we want to have a model that performs well (low E_{train}), but generalises (low E_{test}).

Now let us turn our attention to unsupervised learning. In unsupervised learning we do not have access to a known feature to predict in the training data set. Instead, we want to learn probability distributions of data to solve problems such as generating new data, or grouping data. For instance, in our previous list of features regarding the human body, let us discard the biological sex, age, height and weight and focus on all the other continuous (perimeter) variables. Let us also assume that we have only children and mid-30 adults in our data set. A priori, these two groups of people have very different features because of the changes that the human body undergoes through puberty. In unsupervised learning we would learn the probability distribution of the data set and use this compact representation of the probability distribution to, for example, group individuals into (hopefully) children and adults. We do not know a priori if an individual is a children or an adult. It can be possible, however, that if we find two clear groups, they do not relate to children and adults, but maybe to regular weight people and overweight people. From the lack of knowledge about the error that we are making, since we are not able to quantify prediction errors, a much more difficult approach to the concept of generalisation is laid out. Let us consider an extremely simple case to explain how difficult generalising is in unsupervised learning. A coin is flipped 100 times, yielding 53 heads and 47 tails. We want to calculate the probability that the coin is going to yield heads in flip 101. A frequentist will most certainly predict heads with 53% probability. A more conservative Bayesian would find the results consistent with a fair coin, and would (maybe, let us just imagine...) predict heads with 50% probability. Now, we operate and make predictions only with the knowledge we have, but nothing prevents the results of flipping the coin to be tails 47 times and then heads 53 times for a starter. Thus, it is obvious that this coin is not a fair coin, but a weird coin–weird with respect to our human experience standard.

What lies underneath this reflection on the coin experiment is that different realities can lead to the same results and that we are ignorant of those realities because all that we have is partial knowledge derived from certain measurements. Thus, if we wanted to generalise and predict the result of the next flip of the coin–or, in other words, if we wanted to model the probability distribution of the coin flips we must *assume something* about reality. This point is one of the main differences between the physical sciences and the data sciences: in physics we believe (yes, *believe!*) in natural laws. We believe that God (or whatever metaphysical entity) itself is a mathematician, and we believe that God wrote some equations and made the Universe in their image. In Stephen Hawking's words: what is it that breathes fire into the equations and makes a Universe for them to describe? The last centuries have shown that the induction process that physicists follow through their lives holds true for most of the phenomena that we see, as if, in reality, there was a God that wrote the equations of quantum mechanics and what not. Natural physical laws, then, are our belief that Nature behaves following these universal laws. Indeed, experiments are consistent with these laws on a daily basis. However, experiments in data science come from processes much more complicated (not from first principles) than controlled physical experiments. Thus, data scientists have to make leaps of faith that are incorrect more often than not. But as Kierkegaard said–in other context–one must take leaps of faith and deal with the consequences to become a true individual, so does the scientific community have to take (methodic) leaps of faith for science to advance.

Anyhow, unsupervised learning is different in nature to supervised learning, because it deals with learning probability distributions. A Bayesian look of the world, however, brings supervised learning to the language of probability distributions in a simple way [Bishop, 2006]. A prediction can be seen simply as

$$p_{\boldsymbol{\theta}}(y|x), \tag{4.2}$$

which can be obtained, for instance, through Bayes rule

$$p_{\boldsymbol{\theta}}(y|x) = \frac{p_{\boldsymbol{\varphi}}(x,y)}{p_{\boldsymbol{\vartheta}}(x)}, \tag{4.3}$$

where we have split $\boldsymbol{\theta}$ into $\boldsymbol{\varphi}$ and $\boldsymbol{\vartheta}$. This leverages supervised learning to the language of probabilities, which will come in handy for the framework that we propose, which deals all the time with probability distributions described by density matrices.

Finally, let us give a short explanation on a third branch of machine learning, which is usually regarded as different from either supervised and unsupervised learning: reinforcement learning. Reinforcement learning consists of a learning paradigm whereby an agent can sense its environment, and will take decisions from a subset of previously established automaton decisions given its interaction with the environment. After a series of decisions, the agent will be rewarded or punished.

As an example, let us consider AlphaZero, which is a reinforcement learning algorithm that learns how to play complete-information board games such as chess and Go [Silver et al., 2018]. Complete-information or perfect-information just means that these two-player games provide the whole information to either player, as they both have access to all the information there is to play: the position on the board. In chess, AlphaZero consistently beat the best known chess engine (Stockfish[1]), and most impressively, it was the first Go computer engine to beat the world Go champion, a feat regarded as one of the most important milestones of artificial intelligence. Through Monte Carlo tree search, AlphaZero searched the space of possible moves using a deep convolutional neural network [LeCun et al., 1989; Lecun et al., 1998] for predicting the probability that a move will result in a win/draw/loss for the artificial player.

Again, probability distributions are parameterised by complex machine learning algorithms that model the probability of winning given a chess or Go move. In the end, every machine learning task can be seen from Bayes' view of the world, in which we discover underlying probability distributions from which phenomena arise as the outcome of experiments. To reiterate, in the physical sciences we believe in first principles and natural laws that the outcome of physical experiments must follow. But, in the general look of data science (not only data from physical experiments), we assume that the models that we build generalise sufficiently well to unseen data. The fact that we treat the problem of learning as the problem of estimating probability distributions, joint with the fact that quantum mechanics is a probabilistic theory for the state of physical systems makes the marriage between quantum mechanics (as a mathematical framework) and machine learning, at least, interesting; and as we will see in this chapter, it is useful not only conceptually but in terms of having very good machine learning models.

[1] https://stockfishchess.org/

4.2 QUANTUM MEASUREMENTS FOR SUPERVISED LEARNING

As mentioned before, supervised learning consists of discovering patterns in a set of features in order to predict another feature (a dependent variable) using a training data set. In other words, we try to find complicated correlations (if they exist) between features, and then we use those correlations to predict the dependent variable. We exploit the fact that quantum states allow us to capture probability distributions and correlations between parts of the quantum mechanical system under consideration to *prepare* a quantum state that holds the information of the training data set. I will introduce our optimisation-free approach first, as it is more intuitive, and later on we will move on to the general gradient-based approach.

Building or preparing the training data set quantum state is easy. In Ref. [González et al., 2021b], we explore three different ways of doing this but in this thesis I will focus on only two strategies: pure and mixed states. Before jumping in, let us formally define the problem that we want to solve. We kick off with the problem of classification, that is, we want to have a function $f_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathbb{K}$, where $\mathbb{K} = \{1, 2, \ldots, K\}$ is a set of K labels or K classes. The vector of input features consists of d components, or d dimensions. We have at our disposal a training data set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ that can be split to have a testing data set, where $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{K}$.

We focus on modelling the joint probability distribution of features and labels $p(\boldsymbol{x}, y)$, and later on I will explain how we use this to actually make predictions. First things first, we need a way to map classical data to the quantum state of some system. This is achieved with the so-called quantum feature maps [Schuld and Killoran, 2019] (the name comes from kernel theory [Bishop, 2006]), which can be written as follows. The quantum feature map for the feature vectors $\boldsymbol{x}$ can be defined as the function

$$\begin{aligned} \psi_{\mathcal{X}} : \mathbb{R}^d &\to \mathcal{H}_{\mathcal{X}} \\ \boldsymbol{x} &\mapsto |\psi_{\mathcal{X}}(\boldsymbol{x})\rangle , \end{aligned} \tag{4.4}$$

where $\mathcal{H}_{\mathcal{X}}$ is the Hilbert space of some physical system. Similarly, we define a quantum feature map for the labels

$$\begin{aligned} \psi_{\mathcal{Y}} : \mathbb{K} &\to \mathcal{H}_{\mathcal{Y}} \\ y &\mapsto |\psi_{\mathcal{Y}}(y)\rangle , \end{aligned} \tag{4.5}$$

where, again, $\mathcal{H}_{\mathcal{Y}}$ is the Hilbert space of some other physical system.

With these quantum feature maps, we can map a sample point of the data set with the complete quantum feature map

$$\begin{aligned} \psi : \mathbb{R}^d \times \mathbb{K} &\to \mathcal{H}_{\mathcal{X}} \times \mathcal{H}_{\mathcal{Y}} \\ (\boldsymbol{x}, y) &\mapsto |\psi_{\mathcal{X}}(\boldsymbol{x})\rangle \otimes |\psi_{\mathcal{Y}}(y)\rangle . \end{aligned} \tag{4.6}$$

Quantum feature maps will, hopefully, place data into different regions of the feature Hilbert space $\mathcal{H}_{\mathcal{X}}$ as shown in fig. 4.2. Essentially, we look for quantum feature maps that are able to separate the Hilbert space in a simple way, such that the *decision boundary* between different classes is as easy as possible.

Let us build the training data set using the two strategies mentioned before. A pure data set state is

$$|\Psi\rangle = \mathcal{N}^{-1} \sum_{i=1}^{N} |\psi_{\mathcal{X}}(\boldsymbol{x}_i)\rangle \otimes |\psi_{\mathcal{Y}}(y_i)\rangle , \tag{4.7}$$

where $\mathcal{N}$ is a normalisation constant. The data set quantum state is simple to understand: it is just a superposition of each and every quantum state of every data point.

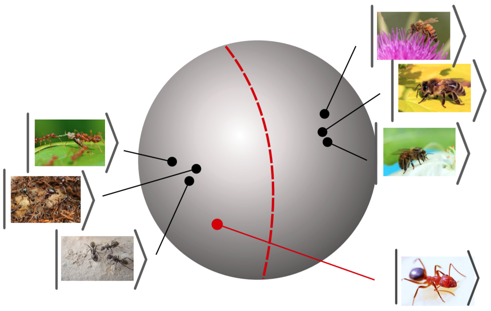

Figure 4.2: Representation of how quantum feature maps take classical data to the Hilbert space of a physical system. Images of ants and bees are depicted as populating different places of the Hilbert space. Such separation (exemplified by the red dashed line, the decision boundary) will later on be used to perform classification. In red, a new ant picture with unknown label (but will be classified as an ant) is shown. Taken from Ref. [Lloyd et al., 2020].

On the other hand, a mixed data set state is

$$\rho = \mathcal{N}^{-1} \sum_{i=1}^{N} |\psi_{\mathcal{X}}(\boldsymbol{x}_i)\rangle\langle\psi_{\mathcal{X}}(\boldsymbol{x}_i)| \otimes |\psi_{\mathcal{Y}}(y_i)\rangle\langle\psi_{\mathcal{Y}}(y_i)| \,, \tag{4.8}$$

where, again, $\mathcal{N}^{-1}$ stands for a normalisation constant.

Correlations captured by eqs. (4.7) and (4.8) are quite different because eq. (4.7) expresses only quantum uncertainty, whereas eq. (4.8) can also express classical uncertainty [Luo, 2005]. The diagram flow for preparing the training quantum state is given in fig. 4.3.

The preparation of the training quantum state yields a quantum mechanical object (a state) that contains the correlations of the data set, as mentioned, in the correlations of the respective quantum state. Of course, the quantum feature map plays a central role in this framework, as it promotes classical data to "quantum data", which is just a quantum mechanical state. Before worrying ourselves with what the quantum feature map should be, let us introduce the protocol for classifying a new data point $\boldsymbol{x}^\star$. The new set of features $\boldsymbol{x}^\star$ will have a corresponding quantum state given simply by the quantum feature map $|\psi_{\mathcal{X}}(\boldsymbol{x}^\star)\rangle$, or, more generally $\rho^\star = |\psi_{\mathcal{X}}(\boldsymbol{x}^\star)\rangle\langle\psi_{\mathcal{X}}(\boldsymbol{x}^\star)|$.

Predicting the class of a new data point is done by projecting the training state ρ_{train} onto the state of the new data point. This operation, which is a projective quantum measurement, results in a collapse of the training state onto the new data point state. In terms of probability, we are collapsing the probability distribution

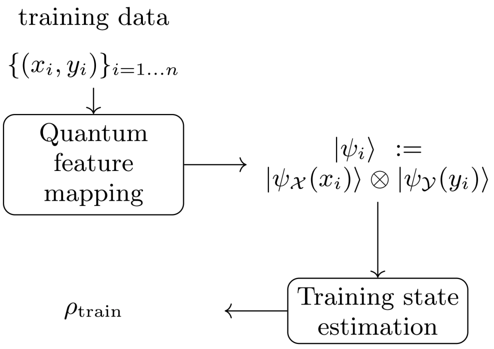
training data
$\{(x_i, y_i)\}_{i=1\ldots n}$
Quantum feature mapping
$|\psi_i\rangle := |\psi_{\mathcal{X}}(x_i)\rangle \otimes |\psi_{\mathcal{Y}}(y_i)\rangle$
Training state estimation
ρ_{train}

Figure 4.3: Diagram flow of the preparation of the training state, where training samples are represented as quantum states of a bipartite system. ρ_{train} can be either given by the pure state density matrix $|\Psi\rangle\langle\Psi|$ in eq. (4.7) or by the mixed density matrix itself in eq. (4.8). Taken from Ref. [González et al., 2021b].

of the training state onto a marginalised probability distribution, because we have essentially got rid of any uncertainty about the state of the train state in the $\mathcal{X}$ part: we know for sure, because of the state collapse, that the state must be completely compatible with the new data point.

We formalise this operation by creating the projector

$$\pi^\star \equiv \pi(\boldsymbol{x}^\star) = |\psi_{\mathcal{X}}(\boldsymbol{x}^\star)\rangle\langle\psi_{\mathcal{X}}(\boldsymbol{x}^\star)| \otimes \mathrm{Id}_{\mathcal{Y}}, \tag{4.9}$$

which projects only the $\mathcal{X}$ part of the system, whereas the $\mathcal{Y}$ part is left untouched due to the identity operator $\mathrm{Id}_{\mathcal{Y}}$.

Now, the projection that takes place is

$$\rho' = \frac{\pi^\star \rho_{\text{train}} \pi^\star}{\mathrm{Tr}[\pi^\star \rho_{\text{train}} \pi^\star]}, \tag{4.10}$$

where we have normalised the state adequately. This new object, ρ' contains all the necessary information that correlates the state of the new data point to the classes. In fact, we can access this information by simply tracing out the degrees of freedom of part $\mathcal{X}$. We are left with a reduced density matrix that contains all the information about the classes only.

$$\rho_{\mathcal{Y}} = \mathrm{Tr}_{\mathcal{X}}[\rho']. \tag{4.11}$$

This whole process can be seen in fig. 4.4.

The reduced density matrix will have, in its diagonal, the probabilities that the new data point corresponds to each class in $\mathbb{K}$. More precisely,

$$p(y = k|\boldsymbol{x}^\star) = \langle\psi_{\mathcal{Y}}(k)|\rho'_{\mathcal{Y}}|\psi_{\mathcal{Y}}(k)\rangle\,. \tag{4.12}$$

Our framework is potent in many ways, as we will show throughout the chapter. But, there are two results which are outstanding for the mixed state choice. I will mention them next, but you can find the formal proof in Ref. [González et al., 2021b]. If the quantum feature maps are one-hot maps (for example the class k is assigned the state $|k\rangle$, and similarly for features, which are assumed to be ordinal), then, our framework is equivalent to Bayesian inference, i.e.,

$$p(y = k|\boldsymbol{x}^\star) = \frac{p(\boldsymbol{x}^\star|y = k)p(y = k)}{p(\boldsymbol{x}^\star)} = \langle k|\rho'_{\mathcal{Y}}|k\rangle\,. \tag{4.13}$$

The second important result is that, regardless of the quantum feature map, our approach is equivalent to a kernel method (see other results on this for general

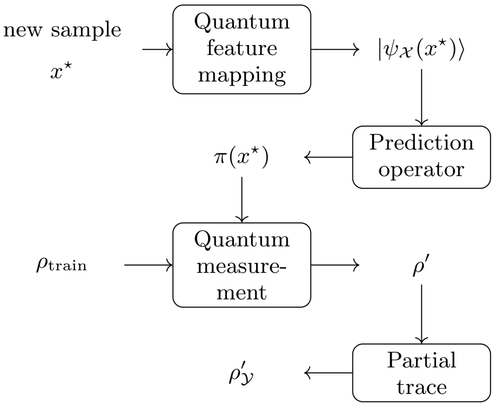

Figure 4.4: Diagram flow of the projection of the training state ρ_{train} onto the new data point state $\rho^\star$. Then, degrees of freedom corresponding to the features are traced out, leaving a reduced density matrix with information about classes only, as in eq. (4.11). Taken from Ref. [González et al., 2021b].

quantum machine learning models defined with quantum circuits in Refs. [Schuld and Petruccione, 2021; Schuld, 2021]), meaning that

$$\rho'_{\mathcal{Y}} = \mathcal{M} \sum_{i=1}^{N} |k(\boldsymbol{x}^{\star}, \boldsymbol{x}_i)|^2 \, |\psi_{\mathcal{Y}}(y_i)\rangle \, \langle\psi_{\mathcal{Y}}(y_i)| \, , \tag{4.14}$$

where $k(\boldsymbol{x}^{\star}, \boldsymbol{x}_i) = \langle\psi_{\mathcal{X}}(\boldsymbol{x}^{\star})|\psi_{\mathcal{X}}(\boldsymbol{x}_i)\rangle$ is the *quantum kernel* and $\mathcal{M}^{-1}$ is just a normalisation constant equal to $\mathrm{Tr}[\pi(\boldsymbol{x}^{\star})\rho_{\mathrm{train}}\pi(\boldsymbol{x}^{\star})]$.

In Ref. [González et al., 2021b] we introduce some quantum feature maps, some of which were seen in the literature before, and we also include a couple of new ones: the softmax quantum feature map, and the random Fourier features quantum feature map, which will come in very handy to prove important generalisation claims that we will make, namely, our framework is able to approximate, to arbitrary precision, any probability distribution. These quantum feature maps are introduced next.

4.2.1 Softmax encoding

We generalise the sine-cosine quantum feature map, which takes variables from $[0, 1]$ and defines the quantum feature map as $\phi : x \mapsto \sin{(2\pi x)} \, |0\rangle + \cos{(2\pi x)} \, |1\rangle$, which creates a superposition in a qubit. The softmax quantum encoding extends this approach from a two-level system (the qubit) to a many-level system (a qudit). In particular, we will use m levels.

First, we define a probability encoding for real values $P : [0, 1] \to \mathbb{R}^m$ via:

$$P_i(x) = \left(\frac{\exp\{-\beta \|x - \alpha_i\|^2\}}{\sum_{i=1}^{m} \exp\{-\beta \|x - \alpha_i\|^2\}} \right)_{i=1 \ldots m} , \tag{4.15}$$

where $\alpha_i = \frac{i-1}{m-1}$. Here, β is a hyper-parameter. Using these probabilities we build a quantum state representing a real number

$$|\varphi(x)\rangle = \sum_{j=1}^{m} \sqrt{P_i(x)} \, |j\rangle \, . \tag{4.16}$$

The quantum state corresponding to a vector $\mathbf{x}_i \in \mathbb{R}^n$ is defined as

$$|\psi_{\mathcal{X}}(\mathbf{x}_i)\rangle = |\varphi(x_{i,1})\rangle \otimes \cdots \otimes |\varphi(x_{i,n})\rangle \, . \tag{4.17}$$

4.2.2 One-hot encoding

This representation corresponds to a basis encoding for discrete variables with m possible values, $\mathcal{X} = \{1, \ldots, m\}$. The encoding for $x = j$ is given by

$$\psi_{\mathcal{X}}(j) = |j\rangle \, . \tag{4.18}$$

4.2.3 Squeezed states

Recently, Schuld and Killoran [2019] proposed to encode data to the phase of a light squeezed state

$$|(r, \varphi)\rangle = \frac{1}{\sqrt{\cosh(r)}} \sum_{n=0}^{\infty} \frac{\sqrt{(2n)!}}{2^n n!} (e^{i(\varphi+\pi)} \tanh(r))^n \, |2n\rangle \, , \tag{4.19}$$

so that a vector $x_i \in [0, \pi]^n$ is mapped to $\psi_{\mathcal{X}}(c, x_i) = |(c, x_{i,1})\rangle \otimes \ldots \otimes |(c, x_{i,n})\rangle$.

4.2.4 Coherent states

Data can also be encoded into the average number of photons of a canonical coherent state [Chatterjee and Yu, 2017]:

$$|(\alpha, \gamma)\rangle = e^{-\frac{\gamma|\alpha|^2}{2}} \sum_{n=0}^{\infty} \frac{\alpha^n \gamma^{n/2}}{\sqrt{n!}} |n\rangle \tag{4.20}$$

where a scaling characterized by γ has been introduced so that the dot product of the two states corresponds to a Gaussian kernel with γ parameter. The mapping from a real data sample $\boldsymbol{x}_j \in \mathbb{R}^n$ to the complex α is done as follows. An auxiliary variable θ_j is built through a min-max scaling of the data set to $[0, \pi]^n$, so that $x_{j,\ell} \mapsto x_{j,\ell} e^{i\theta_{j,\ell}}$. Therefore, a data point $\boldsymbol{x}_j$ is mapped to the quantum feature space through

$$\psi_{\mathcal{X}}(\boldsymbol{x}_j, \gamma) = \left|(x_{j,1} e^{i\theta_{j,1}}, \gamma)\right\rangle \otimes \ldots \otimes \left|(x_{j,n} e^{i\theta_{j,n}}, \gamma)\right\rangle \tag{4.21}$$

which induces a kernel

$$|k_\gamma(\boldsymbol{x}_k, \boldsymbol{x}_j)|^2 = \prod_{\ell=1}^{n} \exp\left(-\gamma |x_{k,\ell} e^{i\theta_{k,\ell}} - x_{j,\ell} e^{i\theta_{j,\ell}}|^2\right), \tag{4.22}$$

where the argument of the exponential is explicitly $-\gamma(x_{k,\ell}^2 + x_{j,\ell}^2 - 2x_{k,\ell}x_{j,\ell}\cos\left(\theta_{k,\ell} - \theta_{j,\ell}\right))$, which imposes a higher distance penalty in the feature space for distant data points in the original space $\mathbb{R}^n$ than the usual Gaussian kernel.

4.2.5 Random Fourier Features

As in quantum state representations, the feature space of kernel methods is a Hilbert space. This means that a quantum feature map implicitly defines a kernel. A natural question is whether the opposite conversion also works, i.e., given a particular kernel function, can we find a quantum feature map to a finite-dimensional Hilbert space such that its inner product corresponds to the kernel? In general, the answer is no; however, it is possible to find an approximation for certain kernels. Random Fourier features (RFF) [Rahimi and Recht, 2008a] provides a technique that finds an explicit finite-dimensional Hilbert space such that the inner product in this space approximates a shift-invariant kernel. Specifically, for a given kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, RFF finds a map $\boldsymbol{z} : \mathbb{R}^d \to \mathbb{R}^D$ such that $k(\boldsymbol{x}, \boldsymbol{y}) \approx \boldsymbol{z}(\boldsymbol{x}) \cdot \boldsymbol{z}(\boldsymbol{y})$.

The quantum state corresponding to a vector $\boldsymbol{x}_i \in \mathbb{R}^d$ is defined as

$$|\psi_{\mathcal{X}}(\boldsymbol{x}_i)\rangle = \frac{1}{||\boldsymbol{z}(\boldsymbol{x}_i)||} \sum_{j=1}^{D} z_j(\boldsymbol{x}_i) |j\rangle . \tag{4.23}$$

We will expand this definition in the future, when we need it, to specify how the map z is constructed. The important point is that we can, for example, approximate a kernel such as the Gaussian kernel with a number of finite terms, as in a Fourier series.

4.2.6 Classification

Each quantum feature map leads to a different kernel in the mixed training state representation of eq. (4.8). We can see the action of previously mentioned quantum feature maps for a classification problem in fig. 4.5.

You can see that, in general, classification is very good for this toy data set. The data set serves as a proof of concept for the framework that we have exposed so far,

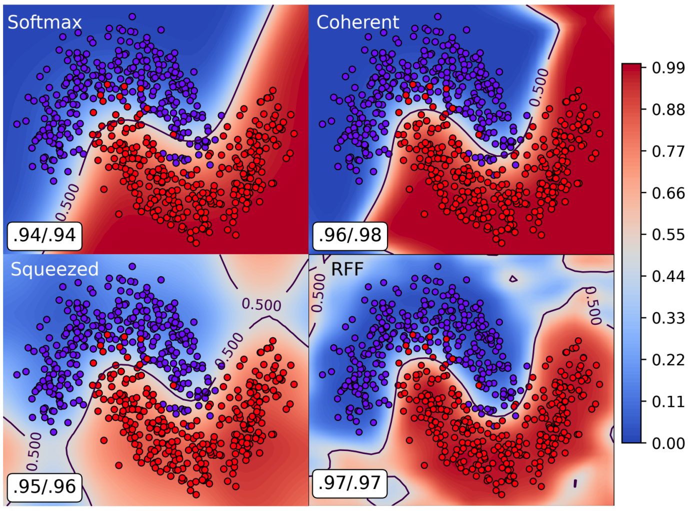

Figure 4.5: Decision heatmap for the two-moons dataset. Panels show classification for the softmax, coherent, squeezed, and random Fourier features states-based feature maps truncated at 20 Fock states. The regularisation parameters (or hyper-parameters) were $\beta = 70$ for the softmax state, $\gamma = 70$ for the coherent state, $r = 2.5$ for the squeezed state, and $\gamma = 20$ for the random Fourier features state. In all four cases, a mixed training state was used. The white boxes show the train/test accuracy of the classifier. Taken from Ref. [González et al., 2021b].

but certainly there are more challenging data sets to perform classification, which we will explore in a while. Before doing that, let us check another toy data set, the spirals data set, by fixing the quantum feature map but varying the approach, with either the pure training data set state in eq. (4.7), the mixed training data set state in eq. (4.8), or a type of state which we call a classical training data set state, which doesn't exploit the correlations that can be induced by the quantum formalism (details about this state can be found in [González et al., 2021b]. Results for classification using the coherent state quantum feature map are shown in fig. 4.6

It is worth emphasising that, for the coherent quantum feature map, which maps classical data to coherent states of light (or bosons, in general), the regularisation parameter γ can greatly impact the quality of classification. Indeed, the parameter γ, as can be seen from eq. (4.20) simply re-scales the number of mean photons of the coherent state. The central part of the figures really show that the mixed training quantum state in eq. (4.8) better captures the relation between classes and (feature) space with respect to the pure training quantum state in eq. (4.7). This is further confirmed when we use other toy data set such as concentric circles with the squeezed-light quantum feature map, shown in fig. 4.7. It is clear that the mixed state is able to better capture the spatial correlation with the class.

Of course we are not restricted to toy data sets, but can look at more challenging ones. In Ref. [González et al., 2021a] we evaluated other data sets which have many more classes, and the dimensionality of the feature space is much greater. In particular, we used the data sets shown in table 4.1. The Letters data set uses 16 different engineered features from hand-written images of the English alphabet [Frey and Slate, 1991]. The USPS is a digit data set of hand-written digits from letters from the United States Postal Service [Hull, 1994]. The Forest data set introduces engineered features to detect different types of forest coverage [Blackard and Dean, 1999]. The MNIST digit data set is probably one of the most famous ones, which includes 28x28 pixels images of digits [Lecun et al., 1998]. The Gisette data set specifically consists of images of the digits 4 and 9, as they are usually confused by classifiers [Guyon et al., 2007]. Finally, we also explore the CIFAR data set which contains tiny coloured images of 32x32 pixels of different objects such as airplanes, dogs, ships, among others [Krizhevsky et al., 2009].

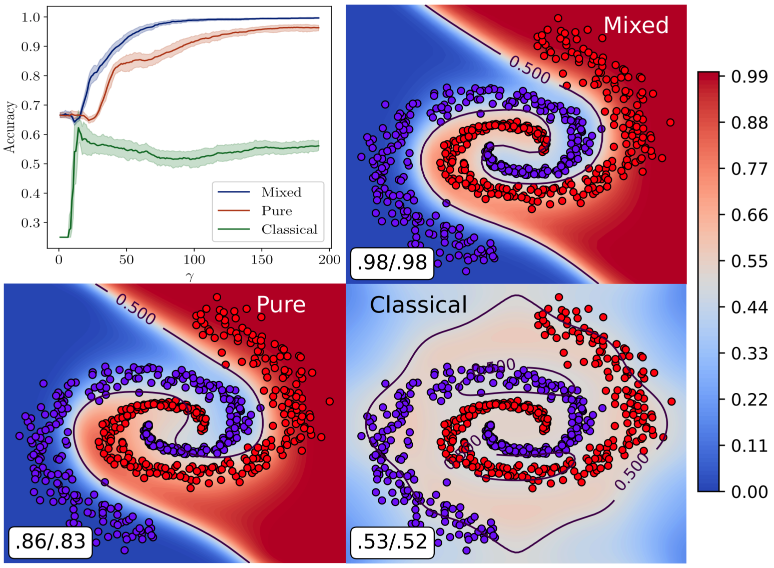

Figure 4.6: Classification accuracy as a function of the regularisation parameter γ on a spirals dataset for mixed, pure and classical training quantum states built with the coherent state encoding truncated at 32 Fock states. Decision heatmaps are shown for the three training schemes at $\gamma = 70$. Figure taken from Ref. [González et al., 2021b].

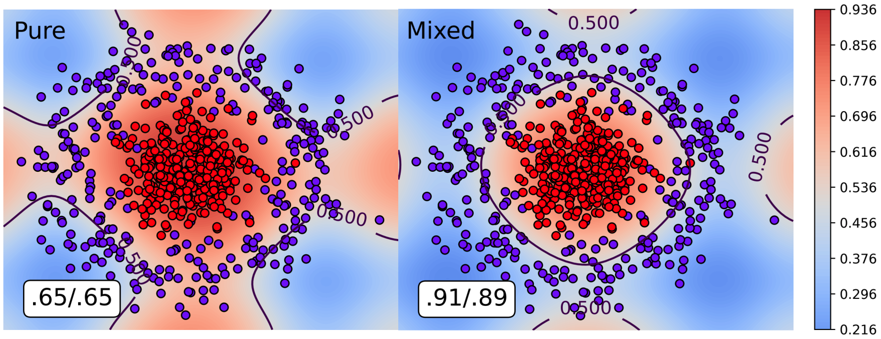

Figure 4.7: Decision heatmap for the concentric circles dataset of the squeezed state quantum feature map with pure and mixed training states. Squeezed states were truncated to the first 10 contributing Fock states, and a value of $r = 2.5$ was used, as in [Schuld and Killoran, 2019]. Taken from Ref. [González et al., 2021b].

Let us start introducing the gradient-based version of the measured classification so far presented, where we ignore the specification of a quantum feature map. Instead, we parameterise the density matrix with a reduced rank factorisation

$$\rho_{\text{train}} = V^T \Lambda V, \tag{4.24}$$

where, in general $V \in \mathbb{C}^{r\times|\mathcal{H}_\mathcal{X}||\mathcal{H}_\mathcal{Y}|}$ and $\Lambda \in \mathbb{C}^{r\times r}$. This approach is equivalent to kernel density classification [Hastie et al., 2009], and therefore receives the name of density matrix kernel density classification (DMKDC), or (DMKDC-SGD) in its stochastic gradient descent (SGD) version, which allows for a learnable density matrix with no explicit quantum feature map. Using the datasets in table 4.1 we compare different methods: a support vector machine over random Fourier features extracted from the data sets (SVM-RFF), the raw DMKDC and DMKDC-SGD algorithms, and also, in the case of explicit images given in the MNIST and CIFAR data sets, we compare the results with the famous LeNet [LeCun et al., 1989] convolutional neural network, and also we use the LeNet neural network as a feature extractor so that features from the LeNet neural network are passed as input to the

Table 4.1: Data sets used for classification evaluation. Taken from Ref. [González et al., 2021a].

Data set	Attributes	Classes	Train-Test
Letters	16	26	14000-6000
Usps	256	10	7291-2007
Forest	54	3	70-30
Mnist	784	10	60000-10000
Gisette	5000	2	4200-1800
Cifar	3072	10	60000-10000

DMKDC algorithm (which is parameter-less). The interested reader is referred to Ref. [González et al., 2021b] for further details. Results are seen in table 4.2.

Table 4.2: Accuracy test results for DMKDC and DMKDC-SGD compared against a linear support vector machine over RFF (SVM-RFF). Two deep learning models are also evaluated on the two image datasets: a convolutional neural network (LeNet) and its combination with DMKDC-SGD (LeNet DMKDC). Taken from Ref. [González et al., 2021a].

Data Set	SVM-RFF	DMKDC	DMKDC-SGD	LeNet	LeNet DMKDC
Letters	0.924±0.0016	0.918±0.0025	**0.9436±0.0021**	-	-
USPS	0.94±0.0012	0.8061±0.0045	**0.9671±0.0018**	-	-
Forest	0.6969±0.0462	0.7272±0.0024	**0.8763±0**	-	-
Gisette	0.9437±0.0032	0.836±0.0035	**0.957±0.0015**	-	-
Mnist	0.9503±0.0062	0.8111±0.0013	0.9516±0.0038	0.989±0.00024	**0.9892±0.00010**
Cifar	0.453±0.0032	0.271±0.005	0.4836±0.0059	0.6162±0.00034	**0.628±0.00015**

4.2.7 Regression

Another supervised learning branch is concerned with regression. So far, we have figured out a way to perform regression using our framework, but regression is restricted to be in a bounded domain. Essentially, we use the same approach of classification, but we separate the bounded space in a grid using the softmax quantum feature map introduced in section 4.2.1.

We place some landmarks at points α_i, which establish a grid (in one dimension) for a continuous variable y, which is the dependent variable. The quantum feature map for the independent variables, or features, can be given or learned. In any case, it can be easily seen that one can estimate the dependent variable to be

$$\hat{y} = E_{\rho_y}[\boldsymbol{\alpha}] = \sum_{i=1}^{m} \langle i|\rho_y|i\rangle\, \alpha_i. \tag{4.25}$$

We highlight that this approach not only allows us to estimate the value of the dependent variable, but also confidence intervals, as our approach is completely probabilistic.

We test our regression framework using ordinal regression data sets, which can be seen as problems where labels not only indicate class membership, but also an order. In table 4.3 we summarise the data sets used to evaluate our regression framework, normally referred as quantum measurement regression (QMR). The diabetes data set consists of glucose measurements in a time-series where events are categorised in different measurements that predict the values obtained in said measurements [Dua and Graff, 2017]. The pyrimidines data set creates the task of predicting the value of the inhibition of dihydrofolate reductase by pyrimidines, where

Table 4.3: Specifications of the data sets used for ordinal regression evaluation. Train and Test indicate the number of samples, which is the same for all the twenty partitions.

Data set	Attributes	Train	Test
Diabetes	2	30	13
Pyrimidines	27	50	24
Triazines	60	100	86
Wisconsin	32	130	64
Machine CPU	6	150	59
Auto MPG	7	200	192
Boston Housing	13	300	206
Stock Domain	9	600	350
Abalone	8	1000	3177

there are three positions in the molecule where chemical activity occurs, and there are nine different attributes per position [King et al., 1995; Larry and Shibin, 2005]. The triazines data set is also related with the geometry of different chemical compounds and relates them to their chemical activity with the so-called quantitative structure-activity relationship for the inhibition of Escherichia coli dihydrofolate reductase rat tumour by triazines [King et al., 1995]. The Wisconsin dataset includes data about breast cancer, where features are characteristics of a fine needle aspirate of a breast mass taken from an image, which characterise the cell nuclei present in that image [Mangasarian et al., 1995]. The Machine CPU dataset contains features of processing cores and also measures the relative performance of a CPU with respect to the vendor's reported performance [Kibler et al., 1989]. The Auto MPG, the data set uses car features to estimate their city-cycle fuel consumption in miles per gallon [Quinlan, 1993]. The famous Boston Housing real estate data set is also used, where one predicts the mean value of houses in a town given socio-economical features of the given town [Harrison and Rubinfeld, 1978]. The stock price data set is a time series data set of daily stock prices from January 1988 through October 1991, for ten aerospace companies [Guvenir and Uysal, 2000]. Finally, the Abalone data set poses the question of predicting the age of abalone from several "easy" physical measurements instead of the time-consuming approach of cutting the shell through the cone, staining it, and counting the number of rings using a microscope [Waugh, 1995].

As can be seen, we used a wide variety of data sets for testing both classification and regression, as shown in tables 4.1 and 4.3. We present the results of regression in table 4.4, where, for comparison reasons, we included an ordinal regression neural network (ORNN), an ordinal extreme learning machine (ORLEM) [Deng et al., 2010], a neural network rank for ordinal regression [Cheng et al., 2008], gaussian processes (GP) and support vector machines (SVM).

We see that in most of the tasks, the learnable version of quantum measurement regression is able to outperform the other methods, while the gradient-free version remains competitive in many cases.

4.3 QUANTUM MEASUREMENTS FOR UNSUPERVISED LEARNING

So far, we have approached supervised learning tasks using the probabilistic view provided by the framework that we introduced mainly in Ref. [González et al., 2021b]. However, since the approach is intrinsically probabilistic, we can easily extend it to estimate probability distributions only, instead of also estimating classes

Table 4.4: Mean absolute error (MAE) test results for QMR, QMR-SGD and different baseline methods: support vector machines (SVM), Gaussian Processes (GP), Neural Network Rank (NNRank), Ordinal Extreme Learning Machines (ORELM) and Ordinal Regression Neural Network (ORNN). The results are the mean and standard deviation of the MAE for the twenty partitions of each dataset. The best result for each data set is in bold face. Taken from Ref. [González et al., 2021a].

Data set	QMR-SGD	QMR	ORNN	ORELM	NNRank	GP	SVM
Diabetes	**0.5115**±**0.067**	0.6115±0.0172	–	0.608	0.546±0.15	0.662±0.14	0.746±0.14
Pyrimidines	0.4083±0.072	0.9458±0.0664	0.677±0.20	0.567	0.450±0.10	**0.392**±**0.07**	0.450±0.11
Triazines	**0.6738**±**0.018**	0.6953±0.0061	–	0.706	0.730±0.07	0.687±0.02	0.698±0.03
Wisconsin	**0.9848**±**0.039**	1.1141±0.0403	–	1.041	–	1.010±0.09	1.003±0.07
Machine	**0.1712**±**0.031**	0.9949±0.0665	0.451±0.03	0.202	0.186±0.04	0.185±0.04	0.192±0.04
Auto	**0.2302**±**0.020**	0.7104±0.0229	–	0.256	0.281±0.02	0.241±0.02	0.260±0.02
Boston	0.2704±0.025	0.6786±0.0183	–	0.322	0.295±0.04	**0.259**±**0.02**	0.267±0.02
Stocks	**0.1029**±**0.011**	0.9714±0.0000	0.127±0.01	0.111	0.127±0.02	0.120±0.02	0.108±0.02
Abalone	0.2329±0.004	0.3069±0.0000	0.635±0.01	0.233	**0.226**±**0.01**	0.232±0.00	0.229±0.00

or dependent continuous (ordinal) variables. I will introduce these ideas a bit more formally than before, because there is a very strong result regarding our ability to approximate arbitrary probability distributions using our framework. Let us kick-off with a complete account of random Fourier features, which extends the ideas of section 4.2.5 by specifying how the random features are actually computed.

4.3.1 Random Fourier Features (Revisited)

Random Fourier features (RFF) [Rahimi and Recht, 2008a] is a method that builds an embedding $\boldsymbol{z} : \mathbb{R}^d \to \mathbb{R}^D$ given a shift-invariant kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ such that $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d, k(x,y) \approx \boldsymbol{z}(\boldsymbol{x}) \cdot \boldsymbol{z}(\boldsymbol{y})$. One of the main applications of RFF is to speedup kernel methods, being data independence one of its advantages.

The RFF method is based on the Bochner's theorem. In layman's terms, Bochner's theorem shows that a shift invariant positive-definite kernel $k(\bullet)$ is the Fourier transform of a probability measure $p(w)$. Rahimi and Recht [2008a] use this result to approximate the kernel function by designing a sample procedure that estimates the integral of the Fourier transform. The first step is to draw D iid samples $\{w_1, \dots w_D\}$ from p and D iid samples $\{b_1, \dots b_D\}$ from a uniform distribution in $[0, 2\pi]$. Then, define:

$$\begin{aligned} &\boldsymbol{z} : \mathbb{R}^d \to \mathbb{R}^D \\ &\quad x \mapsto \sqrt{\frac{2}{D}} \left(\cos\left(w_1^T x + b_1\right), \dots, \cos\left(w_D^T x + b_D\right) \right). \end{aligned} \tag{4.26}$$

Rahimi and Recht [2008a] showed that the expected value of $\boldsymbol{z}(\boldsymbol{x}) \cdot \boldsymbol{z}(\boldsymbol{y})$ uniformly converges to $k(x,y)$:

Theorem 1. (Rahimi and Recht [2008a]) Let $\mathcal{M}$ be a compact subset of $\mathbb{R}^d$ with a diameter $\operatorname{diam}(\mathcal{M})$. Then for the mapping $\boldsymbol{z}$ defined above, we have

$$\begin{aligned} &\Pr\left[\sup_{\boldsymbol{x},\boldsymbol{y} \in \mathcal{M}} |\boldsymbol{z}(\boldsymbol{x}) \cdot \boldsymbol{z}(\boldsymbol{y}) - k(\boldsymbol{x},\boldsymbol{y})| \geq \epsilon \right] \leq \\ &2^8 \left(\frac{\sigma_p \operatorname{diam}(\mathcal{M})}{\epsilon} \right)^2 \exp\left(-\frac{D\epsilon^2}{4(d+2)} \right), \end{aligned} \tag{4.27}$$

where, σ_p^2 is the second momentum of the Fourier transform of the kernel k. In particular, for the Gaussian kernel $\sigma_p^2 = 2d\gamma$, where γ is the spread parameter.

RFF may be used to formulate a non-memory based version of kernel density estimation. For the Gaussian kernel we have an estimation for the density at a point $\boldsymbol{x}$:

$$\begin{aligned}
\hat{f}_\gamma(\boldsymbol{x}) &= \frac{1}{NM_\gamma}\sum_{i=1}^{N} k_\gamma(\boldsymbol{x}_i,\boldsymbol{x}) \\
&\approx \frac{1}{NM_\gamma}\sum_{i=1}^{N}\langle z(\boldsymbol{x}_i), z(\boldsymbol{x})\rangle \\
&= \frac{1}{M_\gamma}\left\langle \frac{1}{N}\sum_{i=1}^{N} z(\boldsymbol{x}_i), z(\boldsymbol{x})\right\rangle \\
&= \frac{1}{M_\gamma}\langle \boldsymbol{\Phi}_{\text{train}}, z(\boldsymbol{x})\rangle \\
&= \frac{1}{M_\gamma}\boldsymbol{\Phi}_{\text{train}}\cdot z(\boldsymbol{x}),
\end{aligned} \tag{4.28}$$

where M_γ is a normalisation constant. $\boldsymbol{\Phi}_{\text{train}}$ in eq. (4.28) can be efficiently calculated during training time, since it is just an average of the RFF embeddings of the training samples. The time complexity of prediction, eq. (4.28), is constant on the size of the training dataset. The price of this efficiency improvement is a loss in precision, since we are using an approximation of the Gaussian kernel.

4.3.2 Probability Density Estimation

Now, we turn our attention to how to estimate the probability density using the previously introduced density matrices composed of quantum feature maps. Of course, we focus on the random Fourier feature quantum feature map in eq. (4.23). We tweak eq. (4.28) in the following manner using the fact that the density matrix predictions are equivalent to a kernel method:

$$\begin{aligned}
\hat{f}_\gamma(x) &= \frac{1}{NM_\gamma}\sum_{i=1}^{N} k_\gamma(\boldsymbol{x}_i,\boldsymbol{x}) \\
&= \frac{1}{NM_\gamma}\sum_{i=1}^{N} k^2_{\gamma/2}(\boldsymbol{x}_i,\boldsymbol{x}) \\
&\approx \frac{1}{NM_\gamma}\sum_{i=1}^{N}\langle z(\boldsymbol{x}_i), z(\boldsymbol{x})\rangle^2 \\
&= \frac{1}{NM_\gamma}\sum_{i=1}^{N}\langle z(\boldsymbol{x}), z(\boldsymbol{x}_i)\rangle\langle z(\boldsymbol{x}_i), z(\boldsymbol{x})\rangle \\
&= \frac{1}{NM_\gamma}\sum_{i=1}^{N}\langle\psi_{\mathcal{X}}(\boldsymbol{x})|\psi_{\mathcal{X}}(\boldsymbol{x}_i)\rangle\,\langle\psi_{\mathcal{X}}(\boldsymbol{x}_i)|\psi_{\mathcal{X}}(\boldsymbol{x})\rangle \\
&= \frac{1}{M_\gamma}\,\langle\psi_{\mathcal{X}}(\boldsymbol{x})|\left(\frac{1}{N}\sum_{i=1}^{N}|\psi_{\mathcal{X}}(\boldsymbol{x}_i)\rangle\langle\psi_{\mathcal{X}}(\boldsymbol{x}_i)|\right)|\psi_{\mathcal{X}}(\boldsymbol{x}\rangle \\
&= \frac{1}{M_\gamma}\,\langle\psi_{\mathcal{X}}(\boldsymbol{x})|\rho_{\text{train}}|\psi_{\mathcal{X}}(\boldsymbol{x})\rangle =: \hat{f}_{\rho_{\text{train}}}(\boldsymbol{x}),
\end{aligned} \tag{4.29}$$

which can also be seen as a projective measurement of ρ_{train} onto the state of the new data point.

With this result, combined with theorem 1 we can show immediately that $\hat{f}_{\rho_{\text{train}}}$, as defined in eq. (4.29), uniformly converges to the Gaussian kernel Parzen's estimator $\hat{f}_\gamma$:

Proposition 1. Let $\mathcal{M}$ be a compact subset of $\mathbb{R}^d$ with a diameter $\text{diam}(\mathcal{M})$, let $X = \{\boldsymbol{x}_i\}_{i=1\ldots N} \subset \mathcal{M}$ a set of iid samples, then $\hat{f}_{\rho_{\text{train}}}$ (eq. (4.29)) and $\hat{f}_\gamma$ satisfy:

$$\Pr\left[\sup_{x\in\mathcal{M}} |\hat{f}_{\rho_{\text{train}}}(\boldsymbol{x}) - \hat{f}_\gamma(\boldsymbol{x})| \geq \epsilon\right] \leq \\ 2^8\left(\frac{\sqrt{2d\gamma}\text{diam}(\mathcal{M})}{3M_\gamma N\epsilon}\right)^2 \exp\left(-\frac{D(3M_\gamma N\epsilon)^2}{4(d+2)}\right). \tag{4.30}$$

The proof of this proposition can be found in the supplementary material of our work [González et al., 2021a]. The $\hat{f}_{\rho_{\text{train}}}$ estimator has an important advantage over the usual Parzen's estimator: its computational complexity. The time to calculate the Parzen's estimator is $O(dN)$ while the time to estimate the density based on the density matrix ρ_{train} is $O(D^2)$, which (remarkably!) is constant on the size of the training dataset.

This last feature is one of the most relevant of our framework: the training data set can be arbitrarily large and the complexity of predicting remains constant. Once the training data set quantum state is estimated, our method is computationally cheap.

Now, we can, again, express the density matrix through a reduced rank factorisation as in eq. (4.24). When this happens, we can avoid building the density matrix to make predictions simply as

$$\hat{f}_{\rho_{\text{train}}}(\boldsymbol{x}) = \frac{1}{M_\gamma}\left\|\Lambda^{\frac{1}{2}}V\left|\psi_{\mathcal{X}}(\boldsymbol{x})\right\rangle\right\|^2, \tag{4.31}$$

where we effectively treat $\Lambda^{\frac{1}{2}}V$ as an operator that can act on the given ket.

For a toy data set of two Gaussian distributions in one dimension, we can see that, as predicted the estimation of the density through density matrix kernel density estimation (DMKDE) is equivalent to kernel density estimation with a Parzen's estimator using the Gaussian kernel. This result is shown in fig. 4.8

Of course, the estimated density also follows the same beaviour in higher-dimensional and more complicated data sets. Another example is a spiral galaxy data set with three arms, as shown in fig. 4.9

Interestingly, we can also learn the random Fourier feature representation. The hyperparameters of the model are the dimension of the RFF representation D, the

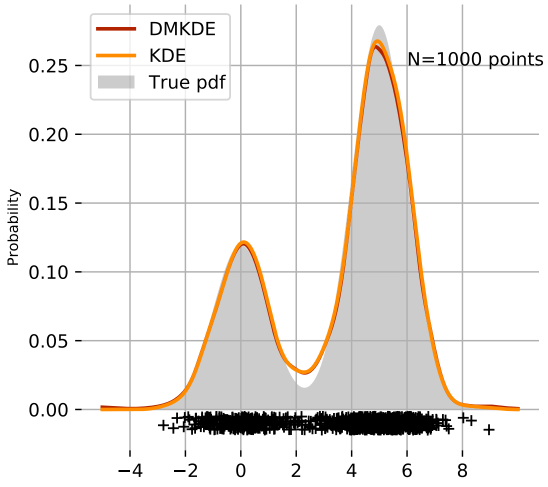

Figure 4.8: 1-D synthetic dataset. The gray zone is the area of the true density. The estimated pdf by DMKDE ($\gamma = 2$) and KDE ($\gamma = 4$) is shown. Taken from Ref. [González et al., 2021a].

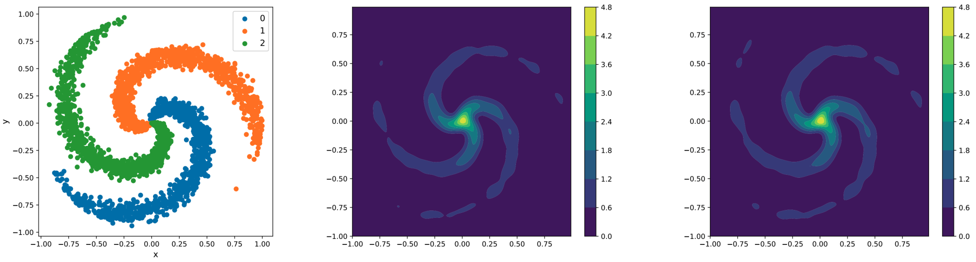

Figure 4.9: 2-D spirals dataset (left) and the density estimation of both KDE (center) and DMKDE (right). Taken from Ref. [González et al., 2021a].

spread parameter γ of the Gaussian kernel and the rank r of the density matrix factorisation. On the other hand, the learnable parameters are the weights and biases of the RFF, $W_{\text{rff}} \in \mathbb{R}^{D\times d}$ and $b_{\text{rff}} \in \mathbb{R}^{d}$, and the components of the factorisation, $V \in \mathbb{C}^{r\times D}$ and $\lambda \in \mathbb{C}^{r}$, the vector with the elements in the diagonal of Λ.

Therefore, since the operations involved in estimating the density of a point $\boldsymbol{x}$ are differentiable, it is possible to use gradient descent to minimise an appropriate loss function, such as the negative log-likelihood:

$$L = -\sum_{i=1}^{K} \log\left(\left\|\Lambda^{\frac{1}{2}} V \left|\psi_{\mathcal{X}}(\boldsymbol{x})\right\rangle\right\|^{2}\right) \tag{4.32}$$

In contrast with the learning procedure based on density matrix estimation, using stochastic gradient descent does not guarantee that we will approximate the real density function. If we train all the parameters, maximising the likelihood becomes an ill-posed problem because of singularities (a Gaussian with arbitrary small variance centered in one training point) Bishop [2006]. Keeping fixed the RFF parameters and optimising the parameters of the density matrix, V and λ, has shown a good experimental performance. The version of the model trained with gradient descent is called density matrix kernel density estimation with stochastic gradient descent, or DMKDE-SGD.

4.4 USING A REAL QUANTUM COMPUTER FOR MACHINE LEARNING

Now, we get to one of the most interesting (and cool) parts of this chapter. So far, I have presented a framework for performing machine learning tasks, both in the supervised and unsupervised setups, based on the mathematical structure of quantum mechanics. We have seen, in particular with proposition 1, that we can approximate arbitrary probability distributions (but this holds not only for the task of kernel density estimation: it extends to the supervised setup, as we are basically estimating the joint density of features and dependent variables). These results mainly hold for the mixed training data set states shown in eq. (4.8).

In this section, I show how we demonstrated that real quantum computers can be used to implement our proposal. However, we do not use the mixed training data set states (eq. (4.8)). We use pure training data set states (eq. (4.7)). The reason behind this is because of practical difficulties. Next, I will give a short explanation about quantum computing (from a computer science perspective), and also about devices (such as the one seen in chapter 2) that are currently being used to build quantum computers.

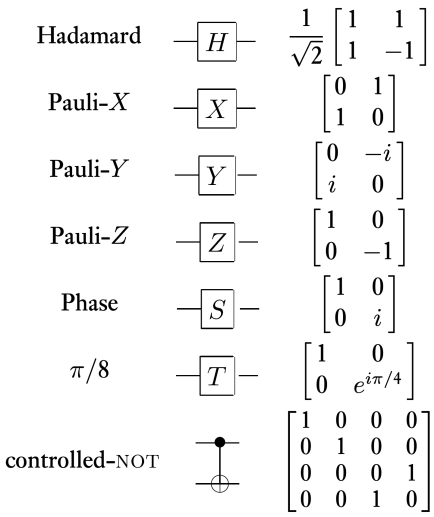
Hadamard
H
$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$
Pauli-X
X
$\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$
Pauli-Y
Y
$\begin{bmatrix}0 & -i\\ i & 0\end{bmatrix}$
Pauli-Z
Z
$\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$
Phase
S
$\begin{bmatrix}1 & 0\\ 0 & i\end{bmatrix}$
$\pi/8$
T
$\begin{bmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{bmatrix}$
controlled-NOT
$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}$

Figure 4.10: Common quantum gates, their quantum circuit diagram representation and their matrix representation, given that $|0\rangle = (1,0)^T$ and $|1\rangle = (0,1)^T$. Taken from Ref. [Nielsen and Chuang, 2002].

4.4.1 Short Introduction to Quantum Computing

Quantum computers are machines that enable the manipulation of the state of a quantum system. We restrict ourselves to quantum computing with qubits, which is the quantum analogue of classical computing with bits. In classical computing, one applies computational gates to bits to change their state. In quantum computing, one applies quantum gates to qubits to change their state. We have already seen in section 2.5 that external fields can be used on a physical device to change the state of a two-level system (a qubit). If one can apply a universal set of quantum gates, one can prepare any state of a system of N qubits. In general, a universal set of quantum gates can be given by a set of single-qubit gates plus the CNOT gate (more on this in a while) [Williams, 2011]. An example of the set of single-qubit gates are the H, S and T gates, whose representation is

$$\begin{aligned} \mathrm{H} &= \frac{1}{\sqrt{2}}\left(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| - |1\rangle\langle 1|\right) && (4.33)\\ \mathrm{S} &= \mathrm{P}(\pi/2) && (4.34)\\ \mathrm{T} &= \mathrm{P}(\pi/4) && (4.35)\\ \mathrm{CNOT} &= |00\rangle\langle 00| + |01\rangle\langle 01| + |11\rangle\langle 10| + |10\rangle\langle 11|\,, && (4.36) \end{aligned}$$

where $|ij\rangle = |i\rangle \otimes |j\rangle$ is a two-qubit state, and

$$\mathrm{P}(\varphi) = |0\rangle\langle 0| + e^{i\varphi}\,|1\rangle\langle 1| \qquad (4.37)$$

is the single-qubit phase gate. In general, quantum gates are just unitary transformations of the quantum state of qubits.

A quantum computer can reset the state of its N qubits to the trivial state $|0\rangle^{\otimes N}$. The action of a gate on a qubit or a set of qubits is usually written in terms of a quantum circuit diagram (like a pentagram for quantum computing). In fig. 4.10 I show some common gates and how they are written inside a quantum circuit.

These quantum gates can be combined in complicated quantum circuits that perform different operations, as shown in fig. 4.11. I stress that the input of the quantum gate can be whatever qubit state. However, in the beginning of a quantum circuit, one normally assumes that one deals with the trivial state $|0\rangle^{\otimes N}$, which one can prepare by "resetting" the quantum computer state.

An important ingredient in quantum computing is not only the ability to manipulate the state of a multi-qubit system. The state is completely inaccessible to us

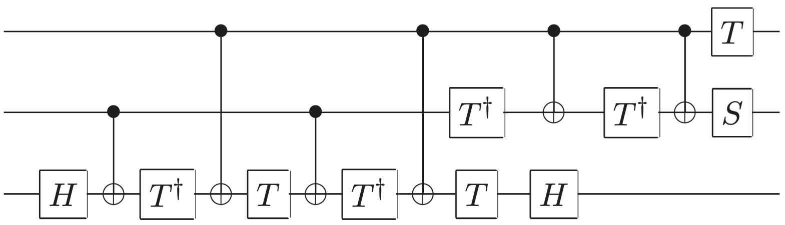

Figure 4.11: Decomposition of a three-qubit gate called the Toffoli gate into one- and two-qubit gates. Taken from Ref. [Nielsen and Chuang, 2002].

if we do not measure. Therefore, we can measure *a la* Stern-Gerlach at the end of the circuit. An example of the measurements "boxes", which represent readout operations, is given in fig. 4.12.

In general, when we measure, we collapse the state of the qubits to the computational basis (the computational basis is a tensor product state of single-qubit states $|0\rangle$ and $|1\rangle$), which can be simply read as classical bit strings. Therefore, in order to extract useful information from the quantum circuit, we need to experimentally realise the quantum circuit many times, with the goal of obtaining statistics from the output. This makes quantum computing clearly different from classical computing: quantum computers are not Turing machines, but probabilistic Turing machines.

4.4.2 Relevance of Quantum Computing

Why is a large part of the scientific community both in academia and the industry pursuing quantum computers for decades? There are many problems in our human existence which are solved computationally. We tend to separate problems by how hard it is to solve them. Problems usually have an intrinsic "size". For example, in the problem "sort these N numbers", the intrinsic size is N. Hard problems are those for which the time that it takes to solve them grows exponentially with respect to their size. Practical solutions to the problems, on the other hand, can be solved in a polynomial time with respect to their size. In reality, we wish to get the solution of the problems, but most of the time we are happy with an approximate solution. Quantum computing not only comes as a hope to getting better approximate solutions exponentially faster than classical computing, but comes as a hope to actually solve problems in polynomial time which can only be solved in exponential time with regular classical computers.

This is all the motivation we need to tackle important problems for our society, as well as quantum mechanics problems (many of which are also very important for human kind, beyond philosophical considerations).

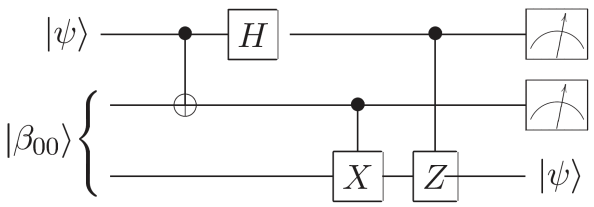

Figure 4.12: Teleportation circuit. Alice owns the first two qubits and Bob owns the last one. Alice wants to teleport the state $|\psi\rangle$ of her first qubit to Bob (notice that $|\psi\rangle$ might be completely unknown to Alice, as it's the only qubit she has with that state, which prevents her to perform quantum tomography). Bob's qubit and Alice's second qubit are initialised in an entangled state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = |\beta_{00}\rangle$. See Ref. [Nielsen and Chuang, 2002] for a complete discussion of the quantum teleportation protocol, from which this figure was taken.

We note that this comparison between classical computers and quantum computers is only an illusion because the two are very different: classical computers are deterministic machines, but quantum computers are probabilistic in their nature. Indeed, classical computation usually puts on the complexity theory table a set of complexity classes (which determine how hard it is to solve problems belonging to such classes), which are different from the complexity classes that quantum computation offers. Probably, the two most famous complexity classes are P and NP, which roughly define the "polynomial"-scaling algorithms and the "exponential"-scaling algorithms, respectively (additionally, solutions to NP problems are easily verified, meaning that polynomial resources are known to be needed to verify a solution, but exponential resources are needed to get a solution). Note that, of course, this distinction is only theoretical and based on our ability to invent algorithms. Usually, people write $P \subseteq NP$, meaning that NP is a more general class than P. We know problems for which no known algorithm is able to solve them in polynomial time, so that we classify the problem as being in NP. But, it can be perfectly possible that, in reality, $P = NP$, which means that we have not discovered efficient algorithms to solve some of the "exponential"-scaling problems. We just do not know.

Our interest is particularly on a complexity class called the bounded-error quantum polynomial time (BQP) complexity class. Bounded-error probabilistic polynomial time (BPP) is the classical counterpart of BQP, which is just the class that can solve a problem with a given probability in polynomial time. Of course, algorithms of quantum computers can only be probabilistic (which is not in detriment to the practical quality of the solutions that one can get with a quantum computer). This means that we (hopefully) create algorithms that have very high lower-bound success rates, so that the probability of failing is very small. So, is quantum computing the panacea? People believe so. A historical break-through milestone is the Deutsch-Jozsa algorithm, which shows that a quantum algorithm can be exponentially faster than any possible deterministic classical algorithm [Deutsch and Jozsa, 1992; Shor, 1999]. The remaining question is, is there a probabilistic classical algorithm which can also be exponentially faster than any possible deterministic classical algorithm? This is the question of whether BPP = BQP. All this discussion can be summarised in a diagram such as the one shown in fig. 4.13.

4.4.3 Decoherence and the Difficulty of Preparing Mixed States

This subsection is devoted to answering the question: why do we implement in a quantum computer our quantum-inspired machine learning framework using the pure training states of eq. (4.7) instead of the more appealing mixed training states of eq. (4.8)? For a starters, we saw in section 4.4.1 that quantum gates are unitary

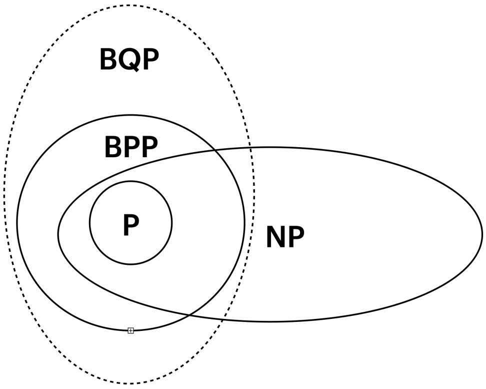

Figure 4.13: Visual summary of the ideas discussed in the main text. BQP is drawn with dashed lines, as we only suspect that it contains BPP and it is the central class of interest. However, all the relations are only a suspicion.

transformations, and, since the initial state of the qubits that compose a quantum computer is the trivial $|0\rangle^{\otimes N}$ tensor product state, which is pure, then the state of the quantum computer will always remain pure. This is in theory. In practice gates are not perfect, and qubits are not perfectly isolated either.

Because qubits are in constant interaction with their environment, the quantum state of the qubit system will be a mixed quantum state. This might be just what we are looking for, but it is not. Our mixed state of eq. (4.8) is a precise one: we need to have complete control over the quantum dissipative channels (each of which is represented by a collapse operator in the GSKL eq. (2.9)) so that we can prepare exactly the mixed state that we want. However, we do not have control over the quantum dissipative channels. We can engineer some quantum dissipative channels (see Ref. [Labouvie et al., 2016] for an example), but not have complete control over them. However, if we had a fault-tolerant quantum computer, i.e., a quantum computer where the state of qubits can be preserved regardless of dissipative channels (because error correction/mitigation methods allow the preservation of such states) and where quantum gates can be applied perfectly, then we could prepare the mixed state in eq. (4.8). An example of how to prepare a mixed state in one qubit is the following, but we require having another ancillary qubit[2]. Consider the state

$$\rho = \frac{1}{d}\sum_{i}^{d} |i\rangle\langle i| \,. \tag{4.38}$$

Then, one can prepare a maximally entangled pure state of two qubits

$$|\phi\rangle \, \frac{1}{\sqrt{d}}\sum_{i}^{d} |i\rangle \otimes |i\rangle \,, \tag{4.39}$$

with an associated density matrix is

$$\varrho = |\phi\rangle\langle\phi| = \sum_{i}^{d}\sum_{j}^{d} |i\rangle\langle j| \otimes |i\rangle\langle j| \,. \tag{4.40}$$

One can verify that tracing the ancillary qubit (the second one, one gets

$$\mathrm{Tr}_2[\varrho] = \rho. \tag{4.41}$$

In current quantum computers from the so-called noisy-intermediate scale quantum computers (NISQ) era, we do not have many qubits available to prepare mixed states, nor we have fault-tolerant quantum computers, meaning that we cannot effectively control the dissipative channels to which qubits are subjected. Therefore, we decided to directly use the pure training states in eq. (4.7) to demonstrate our framework on real quantum computers from IBM, as we show next.

4.4.4 Density Estimation on an IBM Quantum Computer

For estimating probability densities, we will use a simplified notation for the quantum feature map $\psi : \mathcal{X} \to \mathcal{H}_{\mathcal{X}}$, where $\mathcal{X}$ is the space of classical data features, and $\mathcal{H}_{\mathcal{X}}$ is the Hilbert space of some physical system, as we introduced before. The map, therefore, just looks like $\psi : x_i \mapsto |\psi_i\rangle = \psi(x_i)$. Thus, the pure training state for density estimation takes the form

$$|\Psi\rangle = \mathcal{N}^{-1}\sum_{i=1}^{N} |\psi_i\rangle \,. \tag{4.42}$$

[2] This example is taken from https://quantumcomputing.stackexchange.com/a/12539/13291.

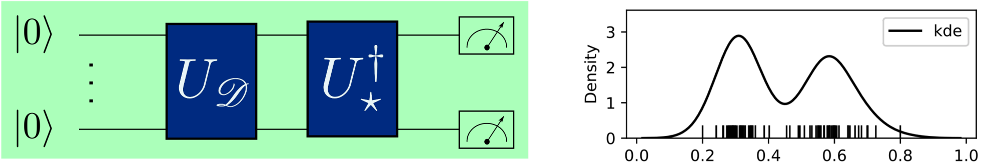

Figure 4.14: Quantum circuit for density estimation, on the left panel, and toy data set used to demonstrate density estimation in a real quantum device, on the right panel. Taken from Ref. [Vargas-Calderón et al., 2022].

For a new data point $\boldsymbol{x}_\star \in \mathcal{X}$, the estimated probability density is

$$\hat{f}(\boldsymbol{x}_\star) = |\langle\Psi|\,\psi(\boldsymbol{x}_\star)\rangle|^2 = |\langle\Psi|\psi_\star\rangle|^2. \tag{4.43}$$

The quantum circuit for probability density estimation is given in the left panel of fig. 4.14. The quantum circuit $U_{\mathcal{D}}$, which acts as a multi-qubit gate, prepares the data set quantum state $|\Psi\rangle = U_{\mathcal{D}}\,|0\rangle^{\otimes N_{\mathcal{X}}}$, where $\mathcal{D}$ is just notation for the training data set; and from right to left, the unitary $U_\star$ prepares the quantum state that corresponds to the point at which we want to estimate the density $|\psi_\star\rangle = U_\star\,|0\rangle^{\otimes N_{\mathcal{X}}}$. Here, $N_{\mathcal{X}}$ is the number of qubits used to represent the data features. Thus, the complete circuit prepares the state $U_\star^\dagger U_{\mathcal{D}}\,|0\rangle^{\otimes N_{\mathcal{X}}}$, whose projection onto $|0\rangle^{\otimes N_{\mathcal{X}}}$ gives the probability density at $\boldsymbol{x}_\star$, i.e., we compute exactly eq. (4.43).

The way in which we actually compute eq. (4.43) is the following: we make M measurements of the quantum circuit and then, we compute $\hat{f}(\boldsymbol{x}_\star) = M_{\mathbf{0}}/M$, where $M_{\mathbf{0}}$ is the number of times that the $\mathbf{0}$ bit string is measured. Explicitly, the complete protocol can be carried out as follows[3] [Vargas-Calderón et al., 2022]:

1. Given a QFM ψ, compute $|\psi_i\rangle = \psi(\boldsymbol{x}_i)$ for each data sample in $\mathcal{D}$.

2. Compute the training data set using eq. (4.42).

3. Use an arbitrary state preparation algorithm to get the circuit $U_{\mathcal{D}}$ that prepares the state in eq. (4.42) on a quantum computer.

4. Compute $\psi(\boldsymbol{x}_\star)$ for a new data point $\boldsymbol{x}_\star$.

5. Use the arbitrary state preparation algorithm in step 3 to get the circuit $U_\star$ that prepares $\psi(\boldsymbol{x}_\star)$.

6. Run the circuit depicted in fig. 4.14 M times to estimate $M_{\mathbf{0}}/M$ with whatever required precision you need.

Note that once the state in eq. (4.42) has been computed, there is no need to perform steps 1-3 to estimate the density of new data points.

Step 3 involves using an arbitrary state preparation algorithm. Most current quantum computers have primitive one- and two-qubit gates that allow universal quantum computation. Therefore, even though the general unitary U_X is known, we need to decompose it into the primitive quantum gates of a quantum computer.

Several algorithms for arbitrary unitary decomposition have been suggested [Barenco et al., 1995; Möttönen et al., 2004; Krol et al., 2022; Li et al., 2013]. In this work, we use the algorithm proposed in Ref. [Shende et al., 2006], that offers a preparation of an n-qubit state using at most $2^{n+1} - 2n$ CNOT gates. This algorithm is implemented in the popular library for quantum computing Qiskit [Treinish et al., 2022], which we used to connect to publicly available quantum computers from IBM.

[3] See https://gitlab.com/ml-physics-unal/qcm for the public library that we released on these circuits, including the protocol in section 4.4.5.

On the right panel of fig. 4.14 you can see a data set that is used to test the density estimation algorithm on a real quantum computer, namely IBM Lima. The quantum feature map used for this task is the random Fourier feature in eq. (4.23):

$$\psi(x) = \sum_{i=0}^{D-1} z_{i+1}(x) \left|\tilde{i}\right\rangle , \tag{4.44}$$

where $\left|\tilde{i}\right\rangle$ is the decimal representation of a bit string of length $\log_2 D$ (in this work, $D = 8$. Thus, $\left|\tilde{0}\right\rangle = \left|0,0,0\right\rangle, \left|\tilde{1}\right\rangle = \left|0,0,1\right\rangle, \ldots, \left|\tilde{7}\right\rangle = \left|1,1,1\right\rangle$. If D is increased, more qubits will be needed, and the depth of the circuit will increase. This will be reflected in a much noisier estimation of the density.). Remarkably, as we proved in [González et al., 2021a], this technique enables the approximation of any probability distribution using finite-dimensional density matrices at the core of the algorithm.

We chose the random Fourier feature quantum feature map to approximate the Gaussian kernel, with a given parameter $\gamma = 80$, such that $z(x) \cdot z(x') \approx e^{-\gamma(x-x')^2}$ [Rahimi and Recht, 2008b]. A total of eight random Fourier features were used so that the quantum circuit in fig. 4.14 consisted of three qubits.

In fig. 4.15 we show the density estimation carried out in three different ways. The three panels correspond to a classical simulation of the density estimation quantum circuit on the left, a classical simulation of the density estimation noisy quantum circuit on the middle (see the next paragraph for an explanation of the noise model), and the density estimation carried out on the IBM Lima quantum device on the right. In the three cases we get a good approximation of the probability density function from which training data was sampled. The discrepancy between the kernel density estimation lines and the quantum circuit ones, even in the ideal case (right panel of fig. 4.15), comes from approximating the Gaussian kernel with a small number of random Fourier features. Finally, we see that the noise model provided by IBM is far from simulating the actual behaviour of the quantum circuit.

The noise model for the quantum circuit, as modelled by IBM's Qiskit [Treinish et al., 2022], simulates imperfect gates that have been fit to experimental measurements to a Krauss noise model [Bogdanov et al., 2013]. It is worth noting that the state of a noisy quantum circuit is never described by a state vector. Instead, it is described by a density matrix. However, this is not exactly how noise is modelled in Qiskit. Imperfect gates are applied instead of the ideal ones with a previously measured probability distribution for selecting the gate to apply. This way of applying noisy gates is analogous to the quantum trajectory approach [Dalibard et al., 1992;

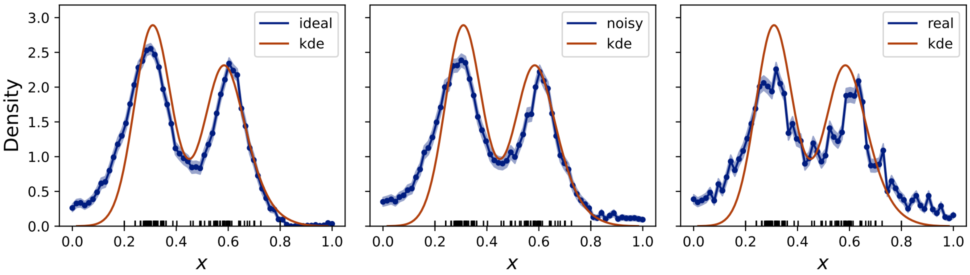

Figure 4.15: Density estimation (blue points) of bi-Gaussian-distributed data with exact circuit simulation (left), noisy circuit simulation (middle, see main text for details on the noise model) and run on the IBM Lima quantum computer. Orange lines are computed through regular Gaussian kernel density estimation. 1024 shots were used to estimate every point on a (simulated or real) quantum computer. Confidence intervals are computed with the asymptotic normal approximation of the Bernoulli distribution from which measurements are sampled. Taken from Ref. [Vargas-Calderón et al., 2022].

Mølmer et al., 1993], where the state vector of the quantum circuit is updated with the application of gates, as we saw in section 2.3. The true quantum state, described by a density matrix, can be recovered by averaging many realisations of the noisy quantum circuit, which is done by running the stochastic quantum circuit many times. The noisy processes that are taken into account are single-qubit readout errors, reset errors, single-qubit Pauli- and $\sqrt{S}$-gate errors and, two-qubit C-NOT gate errors [Chow et al., 2015]. It is clear from the middle and right panels of fig. 4.15 that the used noise model is not able to simulate the real noisy quantum circuit, most likely because such a simplified noise model does not account for the complex dynamics that the quantum circuit undergoes as an open quantum system [Berg et al., 2022].

4.4.5 Classification on an IBM Quantum Computer

Classification makes use of the complete notation that we had in eq. (4.7) for a pure training data set state. Figure 4.16 shows the quantum circuit that performs classification using a labelled training data set $\mathcal{C}$. From left to right, the unitary $U_{\mathcal{C}}$ prepares the quantum state of the data set $\mathcal{C}$: $|\Psi\rangle = U_{\mathcal{C}} |0\rangle^{\otimes N_{\mathcal{X}}+N_{\mathcal{Y}}}$, where $N_{\mathcal{Y}}$ is the number of qubits used to represent the data labels. From right to left, the unitary $U_\star \otimes U_k$ prepares the quantum state of the new data point along the k-th class direction: $|\psi_\star\rangle \otimes |\phi_k\rangle = U_\star \otimes U_k |0\rangle^{\otimes N_{\mathcal{X}}+N_{\mathcal{Y}}}$. Therefore, the quantum circuit prepares a state $(U_\star^\dagger \otimes U_k^\dagger) U_{\mathcal{C}} |0\rangle^{\otimes N_{\mathcal{X}}+N_{\mathcal{Y}}}$ such that its projection onto $|0\rangle^{\otimes N_{\mathcal{X}}+N_{\mathcal{Y}}}$ gives the probability of $\boldsymbol{x}_\star$ being classified in class k, which is completely equivalent to eq. (4.13) when the initial training density matrix is pure.

As in the case of density estimation, the classification probability can be estimated by sampling the quantum circuit M times and counting the number of times $M_{\mathbf{0}}$ that the $\mathbf{0}$ bit string is measured, so that the estimated probability is $\hat{P}(k|\boldsymbol{x}_\star) = M_{\mathbf{0}}/M$.

To summarise, this is the recipe to perform classification:

1. Given a pair of QFMs $\psi_{\mathcal{X}}$ and $\psi_{\mathcal{Y}}$ for data features and labels, compute $\psi_{\mathcal{X}}(\boldsymbol{x}_i)$ and $\psi_{\mathcal{Y}}(y_i)$ for every pair in the classification data set $\mathcal{C}$.

2. Compute the training data set using eq. (4.7).

3. Use an arbitrary state preparation algorithm to get the circuit $U_{\mathcal{C}}$ that prepares the state in eq. (4.7) on a quantum computer.

4. Compute $\psi_{\mathcal{X}}(\boldsymbol{x}_\star)$ for a new data point $\boldsymbol{x}_\star$.

5. Use the arbitrary state preparation algorithm in step 3 to get the circuit $U_\star$ that prepares $\psi_{\mathcal{X}}(\boldsymbol{x}_\star)$.

6. Run the circuit depicted in fig. 4.16, where U_k is not required if $\psi_{\mathcal{Y}}$ is the one-hot encoding (as explained in the appendix of Ref. [Vargas-Calderón et al., 2022].). This circuit has to be run M times to estimate $p(k|\boldsymbol{x}_\star)$ as in eq. (4.13) with a desired precision.

As in the case of density estimation, once the state in eq. (4.7) has been computed, there is no need to perform steps 1–3 to classify a new data point.

The quantum circuit shown in the left panel of fig. 4.16 was used to classify data in a XOR disposition, as shown in the right panel of fig. 4.16. Such toy data set is able to tell apart linear classifiers from non-linear classifiers because it is impossible for a linear classifier to achieve high accuracy. In our case, non-linearity is induced by the quantum feature map. As an example, we consider the cosine-sine quantum feature map:

$$\psi(\boldsymbol{x}) = \psi(x_1, x_2) = \bigotimes_{i=1}^{2} (\sin \pi x_i |0\rangle + \cos \pi x_i |1\rangle), \tag{4.45}$$

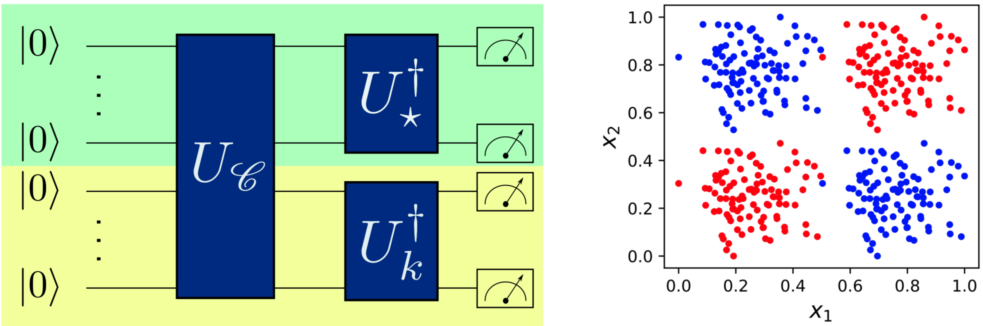

Figure 4.16: Quantum circuit for classification, on the left panel, and toy data set used to demonstrate classification in a real quantum device, on the right panel. Taken from Ref. [Vargas-Calderón et al., 2022].

which ensures that the induced kernel $|\langle\psi(x)|\psi(x')\rangle|^2$ is a pairwise cosine-like similarity measure $\cos\pi(x_1 - x_1')\cos\pi(x_2 - x_2')$. Regarding class labels, we selected the one-hot encoding as the quantum feature map, such that red points are mapped to $|0\rangle$ and blue points are mapped to $|1\rangle$. Thus, a total of three qubits are used to perform the classification quantum circuit, with two qubits encoding the data features, and the remaining one encoding the class label.

Figure 4.17 shows three panels that display the probability that a point placed in $[0,1] \times [0,1]$ is assigned to the red class or the blue class. The three panels correspond to a classical simulation of the classification quantum circuit on the left, a classical simulation of the corresponding noisy quantum circuit on the middle, and the classification carried out on the IBM Bogotá quantum device on the right.

4.5 SUMMARY AND OUTLOOK

This chapter has shown that the mathematical structure exploited in quantum mechanics serves good purpose for machine learning. As a matter of fact, quantum mechanics constantly uses the bridge between geometry and probability laid out by the complex plane (or in general, vectors of complex numbers), which makes it, at the very least, an interesting tool to describe probability distributions which are at the core of machine learning. Our contribution has been to provide a framework that is inspired on the preparation and measurement of physical systems to mimic training and prediction (or data generation), respectively, tasks which cover the

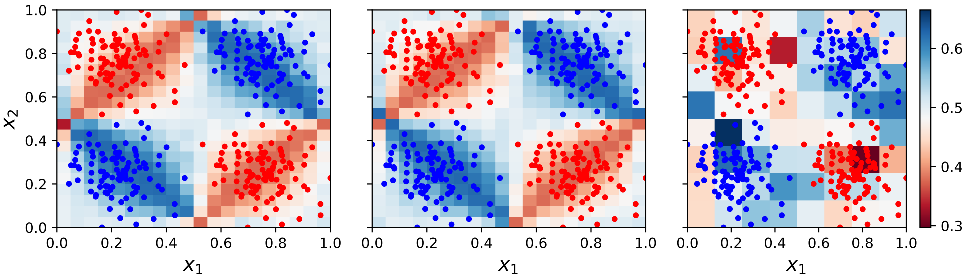

Figure 4.17: Predictions (background colour) of exact circuit simulation (left), noisy circuit simulation (middle) and circuit on the IBM Bogotá quantum device (right) for a XOR data set (points). The colour indicates the probability that a point is classified in the blue class, as shown by the colour bar. The area under the the receiver operating characteristic curve was 99.93%, 99.82% and 95.83% for the predictions of the exact simulation, noisy simulation, and real quantum device, respectively. Taken from Ref. [Vargas-Calderón et al., 2022].

spectrum of supervised and unsupervised learning tasks in the machine learning domain.

Our work has outlined the use of quantum computers (both simulated and real) to perform machine learning tasks using our quantum-inspired framework. This creates a clear pathway of using and studying the physics of quantum devices for clear quantum information processing tasks (chapter 2) which include machine learning. I emphasise that this integration of our quantum-inspired framework with quantum computers is not restricted at all to qubit-based quantum computation. As other students of Fabio's group showed, our framework is (of course!) compatible with qudit-based quantum computation [Useche et al., 2021].

Future investigations will include how to mix the differentiable components of our framework with other classical differentiable components from the field of deep learning in order to perform several machine learning tasks. Of particular interest is multi-modal learning, as our differentiable components are able to condense and represent data from different sources (audio, video, text, tables, etc) in a single format: that of the state of a quantum system.

5 APPLICATIONS TO INDUSTRY PROBLEMS

This chapter is dedicated to applications in the industry, which mainly include combinatorial optimisation problems, but also continuous domain optimisation problems. Such problems are usually NP-hard, meaning that there is no known algorithm that is able to solve them in polynomial time with respect to their characteristic size. Of course we do not provide methods to solve combinatorial problems in polynomial time (this would prove that P = NP), but we use quantum-inspired machine learning techniques to provide algorithms that approximate the solutions of such combinatorial problems.

In section 5.1, a short review is given about the travelling salesman problem and the portfolio optimisation problem, as well as their representation for binary-based solvers. Then, section 5.2 shows how neural quantum states can be used to solve the travelling salesman problem, whereas in section 5.3 a description of generative modelling for optimisation problems is given in the portfolio optimisation problem.

5.1 STATE OF THE ART

In this chapter I will focus on two combinatorial problems. The first one is not particularly interesting for the industry, but tweaking it a bit makes it extremely useful for the industry: the travelling salesman problem. A salesman needs to visit each city, in a given set of cities, exactly once, and return to the city of origin. Of course, this is an interesting problem in operational research, which aims to optimise the time/distance or resources for the salesman in their trip. Routing problems stem from this rather simple, but extremely complicated problem. We propose to solve this problem by mapping it to a Hamiltonian, and then we use neural quantum states to find the ground state of such Hamiltonian. The ground state encodes the solution to the problem. Here, we performed the experiments with the largest Hilbert spaces in this thesis, using systems of almost 100 100-level systems (yes, huge!) [Vargas-Calderón et al., 2021]. The second problem is portfolio optimisation, which aims to, for instance, maximise expected revenue given some willingness to take risks by investing in a portfolio of stock. This problem was tackled by the team at Zapata Computing Inc. before my arrival to the company [Alcazar et al., 2021]. My involvement was in scaling the code for high-performance computing. The centre of this problem is the ability to use generative models that can be classical, quantum-inspired or purely quantum, to propose solutions for the portfolio optimisation problem at hand. The method is called generator-enhanced optimisation, which proposes clever tricks to exploit and explore the solution space of the problem.

The examples shown in this chapter showcase the ability, competition and prospect of quantum computing-related generative modelling to solve real industry problems, which is the business core of the company in which I made my PhD internship, and to which I am engaged nowadays.

5.1.1 Quantum Simulation and the Travelling Salesman Problem

The travelling salesman problem is one of the most studied combinatorial problems. The travelling salesman problem can be mapped to a quadratic unconstrained binary optimisation (QUBO) problem, which is then straight-forwardly mapped to an Ising Hamiltonian [Smelyanskiy et al., 2012; Someya et al., 2016; Minamisawa et al., 2019; Hertz et al., 1991; Kastner, 2005; Warren, 2013; Lucas, 2014]. We use the approach by Smelyanskiy et al. [2012] where binary variables of the form $z_{i\alpha}$ are defined. These binary variables are 1 if the i-th city is the α-th location visited in a tour, and is 0 otherwise.

The length of the tour is

$$\sum_{i,j,\alpha} d_{i,j} z_{i,\alpha} z_{j,\alpha+1}, \tag{5.1}$$

where $d_{i,j}$ is the distance between the i-th an the j-th city. We must also impose that $\sum_i z_{i,\alpha} = 1$ for any α and $\sum_\alpha z_{i,\alpha} = 1$ for any i to ensure that every city is visited exactly once, creating a so-called valid tour. These constraints, however, are only useful conceptually, as they are not of the QUBO form. They can be rewritten as $(\sum_i z_{i,\alpha} - 1)^2 = 0$ for each α and $(\sum_\alpha z_{i,\alpha} - 1)^2$ for each i, so that finding the minimum-length tour of the TSP is equivalent to minimising the quantity

$$\begin{aligned} &\sum_{i,j,\alpha} d_{i,j} z_{i,\alpha} z_{j,\alpha+1} \\ &+ \sum_\alpha \left(\sum_i z_{i,\alpha} - 1 \right)^2 + \sum_i \left(\sum_\alpha z_{i,\alpha} - 1 \right)^2, \end{aligned} \tag{5.2}$$

which is a QUBO problem. It is customary to use penalty coefficients, which multiply the last two terms of eq. (5.2), as explained by Lucas [2014]; Tanahashi et al. [2019]. The reasoning behind adding those penalty coefficients is that the minimisation of eq. (5.2) might get stuck on local minima (with classical optimisation algorithms or quantum annealing setups), and modifying the so-called energy landscape can improve convergence towards the global minimum. However, in our approach we will only sample valid tours, so we will not be worried about the penalty terms at any point.

If we map the binary variable to a spin/qubit σ via $z \mapsto \sigma = 2z - 1$, we obtain the expression of an Ising spin-glass Hamiltonian. Moreover, the ground state of the Hamiltonian is the solution of the TSP, and the corresponding ground energy matches the length of the solution tour by construction. This approach takes N^2 qubits, meaning that the Hilbert space's size is 2^{N^2}. However, in this case we would require $2N^2(N-1)$ spin-spin couplings (each spin is connected to $4(N-1)$ other spins). In fig. 5.1 we depict this situation for a four-city tour example (we change spins by qubits), where the QUBO mapping induces qubit-qubit interactions between qubits that represent a single city (connections between qubits of the same colour) and between qubits that represent different cities (connections between qubits of different colours). The interpretation of these interactions becomes cumbersome. Instead, our proposal in section 5.2 is more naturally related to representing the cities in a tour and their interactions, but also has other benefits which I will present in a while.

5.1.2 Portfolio Optimisation

The other industry application that will be treated here is portfolio optimisation, which is a finance problem that aims to select the best investment strategy with limited resources and a given risk aversion.

Let $\boldsymbol{w}$ be a vector of positive components that add up to one, each of which indicates the proportion of an investment in an asset, meaning that we will invest w_i on asset i. We consider N possible assets. Moreover, we are able to calculate the function of expected return $\langle r(\boldsymbol{w}) \rangle$ and the risk function $\sigma(\boldsymbol{w})$. Given a fixed level of expected return ρ, we aim to minimise the risk, i.e., we will solve

$$\min_{\boldsymbol{w}} \sigma^2(\boldsymbol{w}) |\, \langle r(\boldsymbol{w}) \rangle = \rho, \tag{5.3}$$

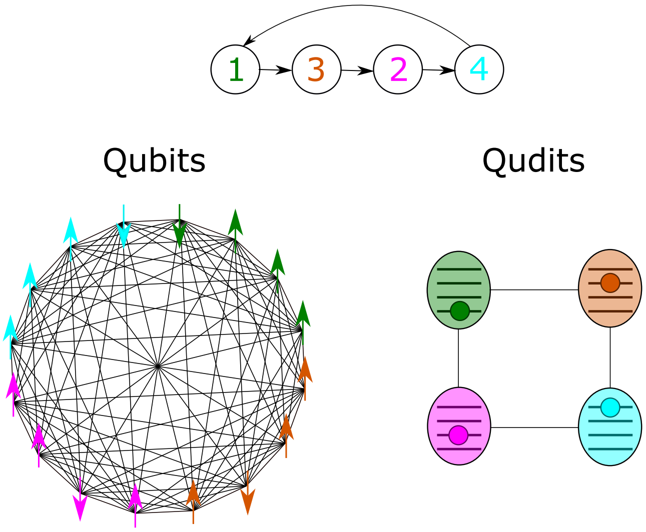

Figure 5.1: Pictorial representation of the TSP map to a system of 4^2 qubits as a QUBO problem and a system of 4 4-level qudits of a toy-example tour with four cities. The top arrow diagram shows a tour starting at city 1. Colours encode the position of a city in the tour. Lines connecting physical sub-systems represent that they are coupled. The binary variables corresponding to each spin can be read in the order $z_{1,1}, z_{2,1}, \ldots, z_{4,1}, z_{1,2}, \ldots, z_{4,2}, \ldots$, starting from the green spin pointing downwards and in clockwise direction. The last index corresponds to the colour or tour position, whereas the first index corresponds to the city label. Figure taken from Ref. [Vargas-Calderón et al., 2021].

where we use the square of the risk function because we compute it as the quadratic form $\boldsymbol{w}^T \cdot \Sigma \cdot \boldsymbol{w}$. The expected return is computed as $\boldsymbol{w} \cdot \boldsymbol{r}$.

In the previous definitions, Σ is the covariance matrix of daily return time series of different assets, and $\boldsymbol{r}$ is the vector of average return for each asset, i.e. in average, how much the price of the asset increases or decreases.

In fact, any quadratic programming solver is very efficient in solving eq. (5.3). However, solving eq. (5.3) yields to a well-known problem in portfolio optimisation problems: over-diversification. This means that one does not put all eggs in one basket: instead, one invests small amounts in many different assets. To avoid this, we impose the so-called cardinality constraint in which we choose only a number κ of assets out of the N total available assets. It means that we need to solve eq. (5.3) with only κ non-zero components in $\boldsymbol{w}$. This is a combinatorial optimisation problem much harder than the previous one, of course.

One can impose other conditions, for instance, that one does not want to invest more than a certain percentage of one's money in a given asset, or invest less than a certain percentage of one's money in the same asset: imposing lower and upper bounds for investment in each asset.

This problem has been historically studied by finance professionals, although it has also caught the attention of statisticians, mathematicians, physicists, and of course engineers. Many different algorithms have been proposed and implemented to solve this problem, but the following are the most relevant. Chang et al. [2000] apply genetic algorithms, tabu search and simulated annealing to portfolio optimisation. Also, improved versions of particle swarm optimisation have been used by Deng et al. [2010], and has been also combined with simulated annealing by Mozafari et al. [2011]. Other population evolution algorithms such as artificial bee colony algorithms [Kalayci et al., 2017, 2020; Cura, 2021] or based on incremental learning have also been proposed for this problem [Lwin and Qu, 2013]. Furthermore, Akbay et al. [2020] use a variable neighbourhood search algorithm combined with quadratic programming solvers. These state of the art results point towards algorithms that are able to explore the feasible solution space with cooper-

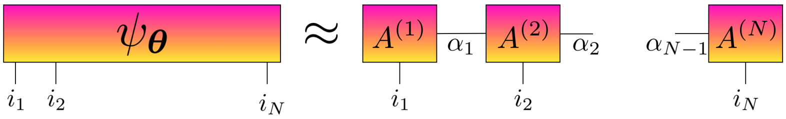

Figure 5.2: Diagram of the matrix product state. The wave function with legs $i_1, \ldots, \text{1}_N$ can be factorised in terms of a tensor train (right hand side of the diagram).

ative agents. In our work, we believe that generative modelling can boost and be on pair (even outperform) these classical proposals.

5.1.3 Tensor Network Generative Modelling: Matrix Product States

The generative model that is mainly used in the Generator-Enhanced Optimisation paper [Alcazar et al., 2021] is the one proposed by [Han et al., 2018]. It consists of one of the most common tensor networks that is a smart factorisation of a tensor consisting of N indices, i.e., an N-rank tensor. The factorisation is

$$\sum_{\alpha_1,\ldots,\alpha_{N-1}} A^{(1)}_{i_1\alpha_1} A^{(2)}_{i_2\alpha_1,\alpha_2} A^{(3)}_{i_3\alpha_2,\alpha^3} \cdots A^{(N-1)}_{i_{N-1}\alpha_{N-2}\alpha_{N-1}} A^{(N)}_{i_N\alpha_{N-1}}, \tag{5.4}$$

which has free physical indices $i_1, \ldots, i_N$. The indices α_j are contracted, and each of them runs from 1 to χ_j, which is called the bond dimension of that specific tensor connection.

In our case, since we will be dealing with binary variables, all the physical indices (also known as legs of the matrix product state) run from 1 to 2, or if we want, from 0 to 1 (just to match binary values).

Again, as in the case of neural quantum states in section 3.1.2, we compute the probability of a bit string $\boldsymbol{b}$ as in quantum mechanics:

$$p(\boldsymbol{b}) = \frac{|\psi_{\boldsymbol{\theta}}(\boldsymbol{b})|^2}{Z}, \tag{5.5}$$

where $\psi_{\boldsymbol{\theta}}$ is the wave function induced by the matrix product state and Z is the partition function. How does the matrix product state induce a wave function? This is a bit cumbersome, but let's start introducing the tensor network diagrams. A tensor can be drawn by an object such as an oval, or a square, with some legs which represent indices. Our particular parameterisation of a matrix product state is shown in fig. 5.2.

With such a diagram, it is easier to explain how to get the wave function for a particular bit string $\boldsymbol{b}$. The bit string has N components which can be all 0 or 1. It's corresponding quantum state, is $|b_1\rangle \otimes |b_2\rangle \otimes \cdots \otimes |b_N\rangle$, which in vector explicit notation can also be looked as another tensor. This is depicted in fig. 5.3. I drew the bit string tensor as separate boxes because even though the bit string is a 2^N-component vector, we can think of it also as a consecutive set of bits. Now, if we have a training data set $\mathcal{T}$, we are just able to adjust the parameters of the model by minimising a cost function. What are the parameters of the model? The tensors A

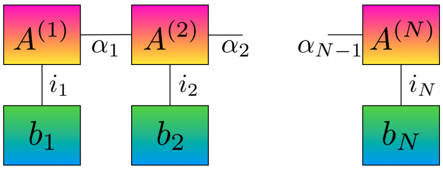

Figure 5.3: Contraction of a matrix product state and a bit string. The physical legs of the matrix product states are contracted with the two-component vectors of each bit in the bit string.

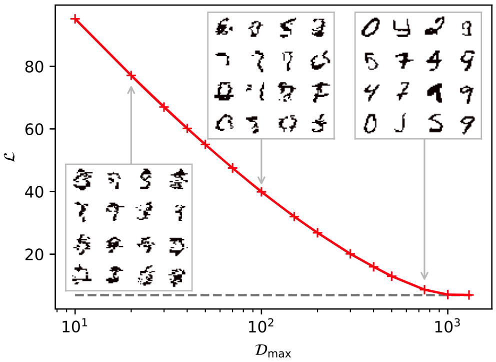

Figure 5.4: Negative log likelihood of an MPS trained with the MNIST digit data set as a function of the maximum bond dimension allowed. Samples are also shown. Figure taken from Ref. [Han et al., 2018].

of the matrix product state have matrix (or rather, tensor) elements, each of which is a free parameter that can be tuned. So, since we maximise the log likelihood, we define the cost function to be the negative log likelihood:

$$\mathcal{L} = -\frac{1}{|\mathcal{T}|}\sum_{\boldsymbol{b}\in\mathcal{T}} \log p(\boldsymbol{b}). \tag{5.6}$$

Han et al. [2018] give an example by training on 1000 random digit images from the MNIST digit data set, shown in fig. 5.4. A maximum bond dimension of $\mathcal{D}_{\max}$ for each contracted index is allowed. We see that as this number is increased the negative lok likelihood lowers, and the quality of generated digit images increase.

5.2 THE TRAVELLING SALESMAN PROBLEM

Given this fact, we only provide a proof of concept of using neural quantum states, and hopefully in the future quantum simulators, to find the solution of this complicated and famous problem. In other words, we did not intend to compete with current solvers, but we make a strong case that neural quantum states (and quantum computing) will be serious candidates for tackling combinatorial problems of the sort. This section presents the results of our work [Vargas-Calderón et al., 2021] and shows some excerpts of it.

Let us now get into our proposal to solve the travelling salesman problem using qudits instead of qubits, as it is customarily done. We propose to use N N-level systems or qudits of N dimensions to map the travelling salesman problem of N cities to the Hamiltonian of a physical system. The corresponding Hilbert space size is $N^N = 2^{N\log_2 N}$, which provides an advantage over the qubit proposal.

Take a look at the four-city example shown in fig. 5.1. We can use four 4-level qudits to encode any tour of four cities. Essentially, the tour, which can be described by a string of consecutive cities to be visited $1 \to 3 \to 2 \to 4 \to 1$, can be encoded in an ordered set of four 4-level qudits, where the first qudit is in the first-level state, the second qudit is in the third-level state, the third qudit is in the second-level state, and the fourth qudit is in the fourth-level state. This is a more natural representation of the tour than the one-hot encoding into qubits produced by the QUBO representation. In fact, from the string representation of the tour, we can assume a quantum analogue problem where the tour is simply depicted as the pure state $|1\rangle \otimes |3\rangle \otimes |2\rangle \otimes |4\rangle$, which is a tensor product of the 4-level occupation of each of the four qudits.

This is easily generalised to a TSP of N cities: let $|n_i\rangle$ be the state of the i-th qudit. In this setup, the i-th qudit occupation refers to the i-th visited city. Therefore, any tour can be represented by a vector $(n_1, n_2, \ldots, n_N)$, where $n_i \neq n_j$ for $i \neq j$, which states that the tour begins at city n_1, then continues to city n_2, and so on, reaching city n_N and then returns to city n_1. As discussed, we assume a quantum analogue problem where the tour vector can be represented as a pure state of N qudits, depicted by a tensor product state of the form $\bigotimes_i |n_i\rangle \equiv |n_1, n_2, \ldots, n_N\rangle \equiv |\boldsymbol{n}\rangle$. This allows us to define the Hamiltonian via its element matrices as

$$\langle \boldsymbol{n}| \hat{H} |\boldsymbol{n}\rangle = \begin{cases} d_{n_1,n_2} + d_{n_2,n_3} + \ldots + d_{n_N,n_1} & \text{if } (n_1, \ldots, n_N) \text{ is a permutation of } (1, 2, \ldots, N), \\ p & \text{otherwise,} \end{cases} \tag{5.7}$$

where $p \gg \max\{d_{i,j}\}$ is a term that penalises configurations that do not correspond to valid tours. Such a penalty term can be compared to an effective exclusion principle, where invalid tours cannot exist. Moreover, $\langle \boldsymbol{n}| \hat{H} |\boldsymbol{n}'\rangle = p$ for $\boldsymbol{n} \neq \boldsymbol{n}'$, which will add energy to the system if the ground state is not a pure tensor state.

A Hamiltonian of this form may arise from a sum of local Hamiltonians (which is helpful if we want to use NetKet to solve it!), which are two-qudit operators, whose matrix elements are

$$\langle i, j| \hat{D} |\ell, m\rangle = d_{i,j}\delta_{i,\ell}\delta_{j,m} + p'(2 - \delta_{i,\ell} - \delta_{j,m}), \tag{5.8}$$

where $\delta_{i,j}$ is the Kronecker delta and $p' \gg \max\{d_{i,j}\}$. Thus, the Hamiltonian of the N-qudits system would be

$$\hat{H} = \hat{D}^{(\mathcal{H}_1 \otimes \mathcal{H}_2)} + \hat{D}^{(\mathcal{H}_2 \otimes \mathcal{H}_3)} + \ldots + \hat{D}^{(\mathcal{H}_N \otimes \mathcal{H}_1)}, \tag{5.9}$$

where $\mathcal{H}_i$ is the Hilbert space of the i-th qudit and $\hat{D}^{(\mathcal{H}_i \otimes \mathcal{H}_j)}$ is the operator in eq. (5.8) acting on the space $\mathcal{H}_i \otimes \mathcal{H}_j$. Notice that the Hamiltonian in eq. (5.9) is slightly different than the one presented in eq. (5.7) because the penalty term becomes a collection of penalty terms, depending on how many repeated occupations there are in the state. Again, by construction, any state $|\boldsymbol{n}\rangle$ corresponding to a valid city tour will have an energy equal to the tour distance, which is why minimising the energy yields the ground state of the Hamiltonian in eq. (5.9), which corresponds to the travelling salesman problem solution.

Not only is the Hilbert space of the qudit system much smaller than the qubit system, there is also a decrease in the possible physical implementation of the qudit chip, as the graph connecting different qudits becomes highly sparse. In fact, the form of eq. (5.9) shows a nearest-neighbour Hamiltonian of a 1D system with periodic boundary conditions, which makes it explicit that only N qudit-qudit couplings are needed (in contrast to $\mathcal{O}(N^3)$ qubit-qubit couplings). Thus, in future qudit-based quantum computers, this problem can be solved using qudits arranged in a ring. Also, each qudit must only be connected to a constant number of two other qudits (in contrast to $\mathcal{O}(N)$ for the qubit case).

We performed variational Monte Carlo experiments on two different setups using the restricted Boltzmann machine neural quantum state for the qubits case, and a convolutional neural network quantum state for the qudits case.

The first setup corresponds to the QUBO representation of the travelling salesman problem, mapped to an Ising Hamiltonian, i.e. a qubit Hamiltonian as shown in eq. (5.2). For this representation, we use a restricted Boltzmann machine as the neural quantum state because it naturally takes as input binary variables. The second setup corresponds to our many-qudit representation of the travelling salesman problem as shown in eq. (5.9). In this case, we use a convolutional neural network (CNN) as the neural quantum state because it naturally expresses translational symmetry, which exists in the travelling salesman problem [1].

For each setup, we consider a toy-travelling salesman problem, where cities are placed on a line with coordinates $x_n = n$. This class of travelling salesman problem

[1] It does not matter which city is the first one to be visited, the tour length will be the same if we shift the positions of the tour.

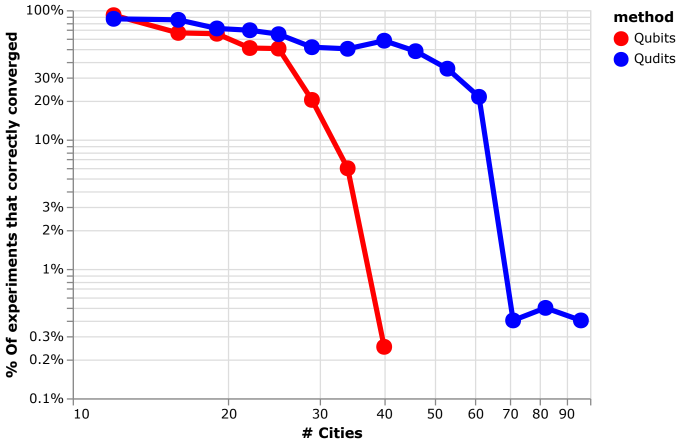

Figure 5.5: Percentage of experiments that converged to the desired solution for the many-qubit (red) versus the many-qudit (blue) representation of the travelling salesman problem, as a function of the number of cities. The lines are shown to guide the eye only. Figure taken from Ref. [Vargas-Calderón et al., 2021].

allows us to quickly benchmark the solutions obtained because the minimum tour length is $2(N-1)$, where N is the total number of cities considered in the city chain. As it has become customary in this thesis, we perform extensive experiments covering a large range of different hyper-parameters with Optuna [Akiba et al., 2019]. In total, we consider 400 sets of hyper-parameters for each of the two setups. The interested reader might find the experimental details in Ref. [Vargas-Calderón et al., 2021].

Figure 5.5 shows the percentage of experiments that correctly converge to the ground energy as a function of the number of cities considered in the linear layout. Interestingly, we see a significant drop in the percentage of experiments that converged to the expected solution around 40 cities for the qubit representation of the travelling salesman problem (notice that this corresponds to a system with 1600 qubits, which is indeed a very challenging problem). In contrast, the drop is located at around 70 cities for the qudit representation. Such a drop indicates how rapidly the travelling salesman problem solvability decreases with the number of cities, exposing its computational hardness. Moreover, we hypothesise that this drop might be connected to a phase transition of the variational Monte Carlo + neural quantum state algorithm when exposed to the travelling salesman problem, as this behaviour has previously been seen in other algorithms for the travelling salesman problem [Gent and Walsh, 1996].

Another essential feature of the experiments carried out is that experiments that converge take less time in the many-qudit representation than in the many-qubit representation. In fig. 5.6 we exemplify this fact for some number of cities, where the convergence of the best (most accurate and fast) experiments of the qubit (orange) and qudit (blue) representations of the TSP are shown. Not only are the best experiments of the qudit representation better than those of the qubit representation (except in the case of 16 and 22 cities[2]), but the difference in performance also grows as the number of cities grows.

Our work paves the way towards the exploitation of future qudit-based quantum computers for solving combinatorial problems such as the famous NP-complete travelling salesman problem, whose solution is based on a many-qudit Hamiltonian that could be implemented in such computers.

[2] Note that the experiments shown in fig. 5.6 are those that are in a local minimum of the hyper-parameter space.

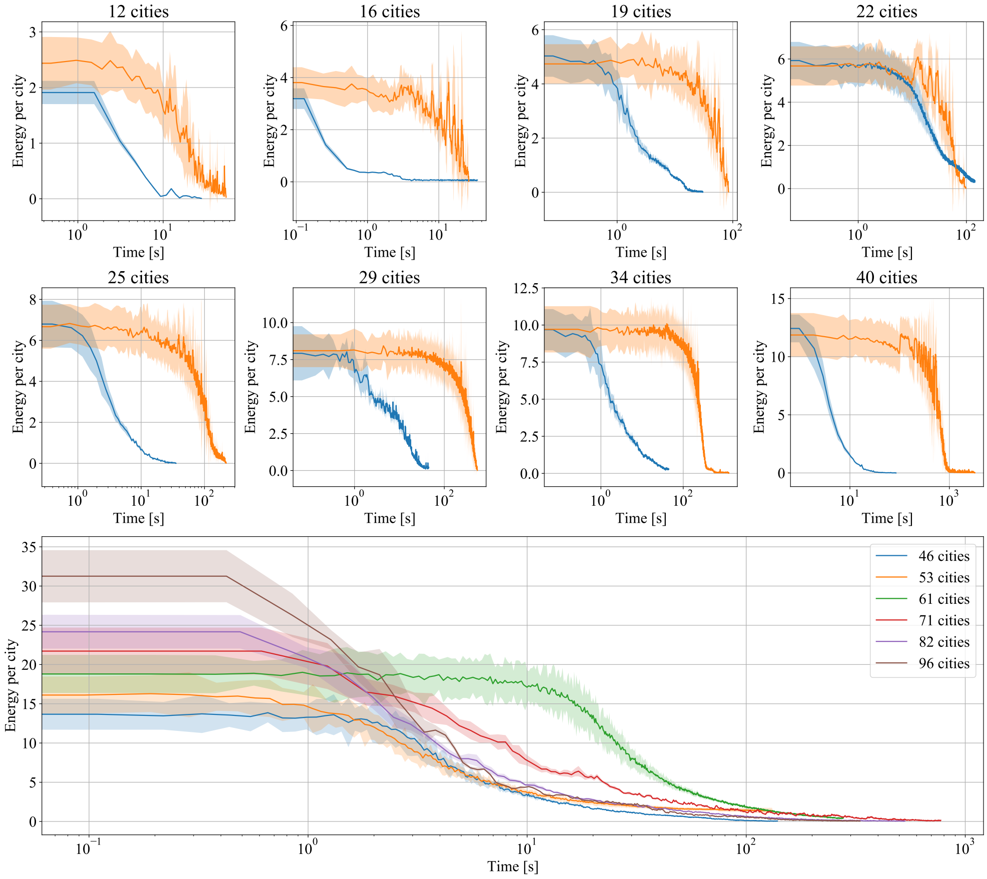

Figure 5.6: Energy convergence as a function of processing time in seconds. The first two rows show the energy convergence of the best experiment for the qubit (orange) and the qudit (blue) representations of the travelling salesman problem, with respect to the baseline energy, which is $2(N-1)$ for N cities. The bottom panel shows the energy convergence of the best experiments for the qudit representation of the travelling salesman problem for several other number of cities. In all plots, the shaded areas correspond to 2 standard deviations of the energy of each Metropolis-Hastings sample. Figure taken from Ref. [Vargas-Calderón et al., 2021].

5.3 PORTFOLIO OPTIMISATION THROUGH GENERATIVE MODELLING

As explained in section 4.1.1, machine learning targets the problem of creating compact representations of probability distributions in any given space. In this section I will present one of the main applications of the Quantum Artificial Intelligence team at Zapata Computing Inc.[3], which is used to solve difficult optimisation problems. As I mentioned earlier, I have been part of a team which has leveraged this algorithm for high-performance computing, which is included in a private suite for quantum machine learning, to which Zapata Computing Inc. clients will have access to. In particular, I will dedicate this section to explain the ideas of Ref. [Alcazar et al., 2021].

The proprietary framework called generator-enhanced optimisation (GEO) is a combination of heuristic rules that guide a generative model in order to look for solutions of challenging optimisation problems. The framework is depicted in fig. 5.7.

[3] https://www.zapatacomputing.com/team/vladimir-vargas-calderon/

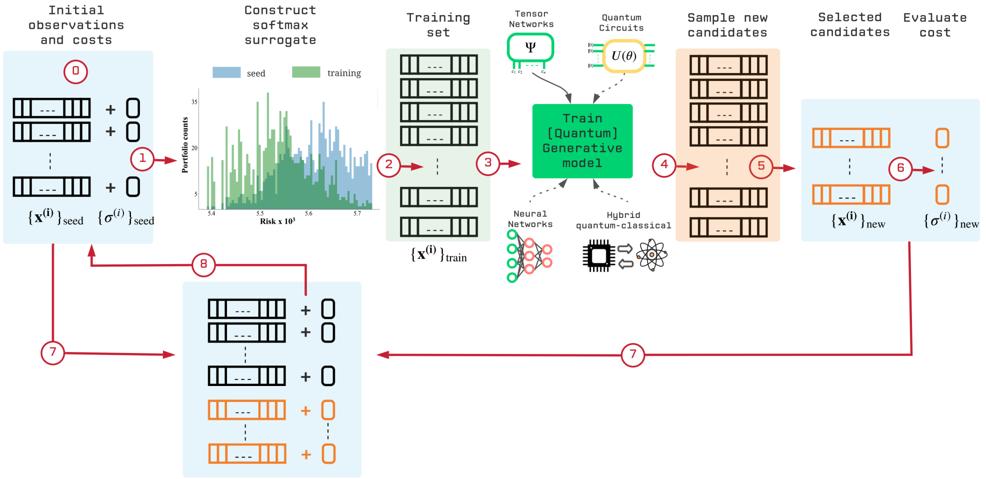

Figure 5.7: Diagram of the generator-enhanced optimisation framework. Explanation is in the main text. Figure taken from Ref.[Alcazar et al., 2021].

Originally, GEO has been applied to combinatorial problems defined on binary variables. That is, we want to find a bit string $\boldsymbol{b} \in \mathbb{B}^N$ of length N that minimises a certain objective function $O : \mathbb{B}^N \to \mathbb{R}$. In step 0 of fig. 5.7 we have optional seed data consisting of bit strings $\boldsymbol{b}$ and their respective objective or cost value $O(\boldsymbol{b})$. Such seed data might come from other classical optimiser, or from some systematic way to initially sample the solution space.

In step 1, there is a construction of a map from cost values to probability values. Basically, we will assume that there is an underlying probability distribution of the bit strings so that observed bit strings with lower cost will have an associated higher probability, and viceversa. This will induce, in the future, the notion of being able to generate new data which will be more "probable", and hopefully will have an even lower cost. For example, if we have a set of costs $\{y_i = O(\boldsymbol{b}_i)\}$, we can use the softmax function to produce a vector of probabilities $\boldsymbol{p}$ such that

$$p_i = \frac{\exp(\beta y_i)}{\sum_j \exp(\beta y_j)}. \tag{5.10}$$

In step two we assign the probabilities to each bit string in the data set. With such data set, one can train a generative model in step 3. These generative models can be classical, such as generative adversarial networks [Goodfellow et al., 2014], restricted Boltzmann machines [Smolensky, 1986], variational autoencoders [Kingma and Welling, 2013], among others; quantum inspired, such as tensor network generative models [Han et al., 2018]; or purely quantum, such as quantum circuit Born machines [Cheng et al., 2018].

After the generative model is trained, one can use it to generate new data in step 4, meaning that one has new candidate bit strings. In step 5, under some heuristic rule, one selects a subset of the generated bit strings so that in step 6, we measure the cost value of each of those post-selected bit strings.

In step 7, the latest data set used to train the generative model is enlarged with the new bit strings and their respective costs to form a new initial data set, shown in step 8. All these steps form a single iteration of GEO. GEO is repeated until one improves on the minimum possible cost that we find.

Now that the GEO strategy has been laid out, I will explain both the example problem of portfolio optimisation and also the tensor network generative model.

5.3.1 Results

GEO can act as a booster or as a stand-alone component. This means that it can take bit strings and cost values explored by other classical solver (booster mode), or it can start from zero knowledge of the cost landscape (stand-alone mode). In the booster mode, Alcazar et al. [2021] used simulated annealing [Pincus, 1970] to get seed data so that GEO could improve upon it by generating new, better bit strings with lower costs. The results are shown in fig. 5.8. We can see that there are outstanding bit strings generated by GEO with lower costs than simulated annealing-generated bit strings. In particular, we find that for the case of a total universe of 50 assets, GEO finds 31 outstanding bit strings, whereas in the case of 100 assets, GEO finds 349 outstanding assets.

In the stand-alone mode, Alcazar et al. [2021] compare the evolution of the minimum observed cost between different classical solvers compared with the tensor-network GEO (TN-GEO). The following solvers are considered. First, a random solver, which fully randomly searches the space of all possible asset configurations (binary search) and then uses quadratic programming to minimise the risk. The conditioned random solver is practically the same as the random solver, but it uses the fact that only 50% of the assets can be used to minimise the risk. Simulated annealing is also considered, as well as Bayesian optimisation using the library GPyOpt [The GPyOpt authors, 2016]. Results show that TN-GEO consistently outperforms the other solvers, as shown in fig. 5.9..

The interested reader is referred to Ref. [Alcazar et al., 2021] where other extensive comparisons against state of the art solvers for the portfolio optimisation problem are shown with different asset indices.

5.4 SUMMARY AND OUTLOOK

This section presented applications of generating data to solve combinatorial optimisation problems. In particular, the travelling salesman problem and the portfolio optimisation problems were explored by using neural quantum states and Zapata's proprietary generator-enhanced optimisation. Quantum-inspired generation of possible solutions has shown outstanding capability in the case of the travelling salesman problem, being able to find solutions for systems of 96 qudits, each with 96 levels, which makes up a huge Hilbert space, as well as solutions of up to 1600

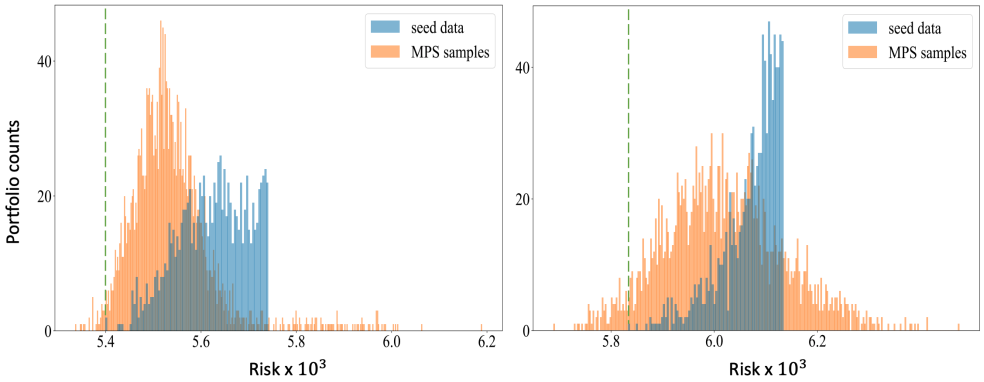

Figure 5.8: Seed data from simulated annealing (blue histogram) is used for GEO as a booster in the cases of $N = 50$ possible assets (left panel) and $N = 100$ possible assets (right panel). Costs obtained with GEO using a matrix product state (MPS) generative model are shown in the orange histograms. The vertical green dashed line indicates the position of the best cost obtained through simulated annealing. The cardinality constraint was set to $N/2$.

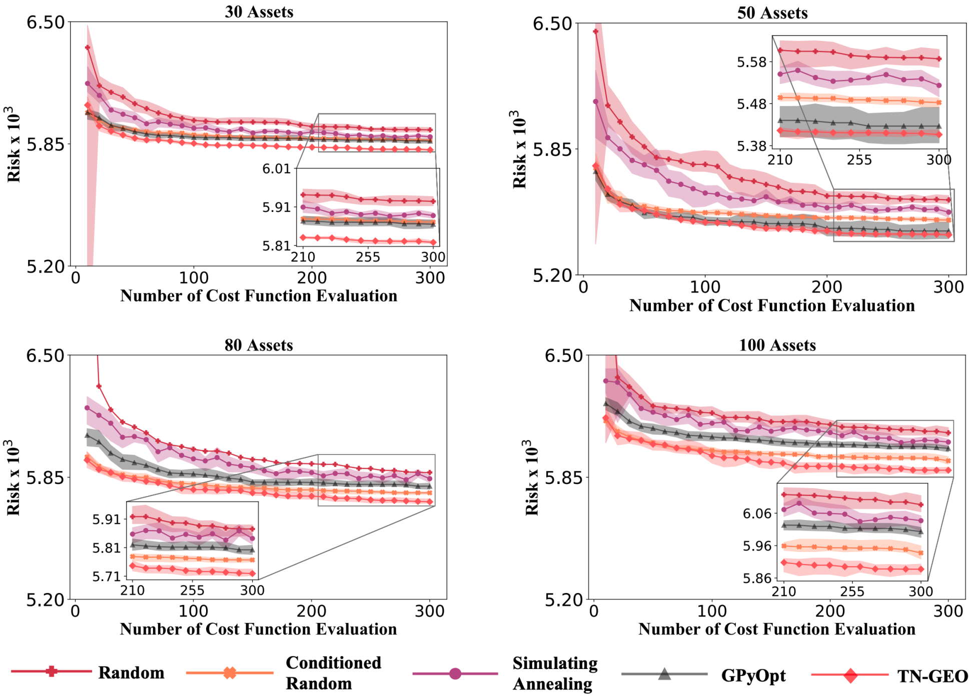

Figure 5.9: Comparison between different classical solvers and the TN-GEO solver, where the former clearly outperforms the rest for different sizes of the asset universe.

qubits. Also, in the case of portfolio optimisation, matrix product states have shown the ability to approximate probability distributions, and to act as generative models that provide generated bit strings which consistently improve upon the bit strings found by other classical solvers in order to find the best strategies for investment given a universe of assets.

So far, the ability to map combinatorial problems to quantum physics problems has shown to be crucial in order to use quantum simulators (and why not quantum computers) to find the solution to such combinatorial problems. This ability is, for the time being, human. Our novel approach to mapping the travelling salesman problem not to the usual quadratic unconstrained binary optimisation problem, but to a problem of sparsely interacting qudits was fruitful. Ways to map different combinatorial problems to physical systems need to be devised (especially for known NP-complete problems, which are completely equivalent to other NP-complete problems) in order to propose experimental platforms to solve them.

Also, exploration of different heuristics and generative models (which we already do at Zapata through adversarial networks with Born machine priors [Rudolph et al., 2022], among other quantum-inspired and quantum generative models) will improve the generator-enhanced optimisation framework to solve industry-scale problems such as the portfolio optimisation one. Their extension to problems defined in the continuous domain is also central for many other problems in the industry [4].

[4] No details are given in this regard as we (Zapata Computing Inc.) are at the moment of writing in patent filing paperwork.

6 CONCLUSIONS

This thesis has mainly explored a two-way avenue in which machine learning and the study of quantum physics coexist and improve each other: the symbiosis of machine learning and quantum physics. The most natural connection between these two fields is quantum machine learning, which studies quantum algorithms to solve machine learning tasks (supervised and unsupervised). However, throughout this thesis, we show that the connection between these two fields goes far beyond, and connects quantum computing, quantum physics theory and machine learning.

So far, quantum computing has been widely concerned with qubit-based computation. However, there is an ever growing community that researches on alternative approaches to using quantum physical systems to perform quantum information processing operations. I have dedicated chapter 2 to the development of one such alternatives: quantum control of many-level systems through giant Rabi oscillations in an optomechanical experimental setup. From the quantum optics perspective, we have shown that the use of novel interfaces between quantum emitters, acoustic cavities and driving lasers, one can precisely control the occupation and interaction between pairs of states. All this work–which is a continuation of my undergraduate work–in combination with the rest of the projects contained in this thesis, point towards the need to further explore physical platforms for quantum simulation and quantum computation. Most likely, the superconductivity and nanotechnology group led by Herbert will continue to study this platform with more advanced control techniques, for instance involving external magnetic fields. The code base that is left for the group[1] to use already has all the interfaces necessary to perform extensive and systematic numerical experiments with an external magnetic field as the one modelled by Jiménez-Orjuela et al. [2017].

Later on, in chapter 5 we showed that qudit-based quantum simulators/computers can greatly simplify the experimental realisation of mapping the travelling salesman problem to a quantum computer. This highlights the importance of exploring quantum control in realistic physical devices (from the theoretical point of view, but of course from the experimental point of view as well). We proposed a novel approach to mapping the travelling salesman problem to the ground state of a sparse many-qudit Hamiltonian. Most remarkably, our approach requires a constant and very sparse connectivity between qudits, whereas its qubit counterpart connectivity scales with the total number of cities under consideration by the travelling salesman problem. From current quantum computing architecture, the qubit alternative is unfeasible, but our proposal could be achieved by placing qudits in a ring setup, where qudits interact only with their two nearest neighbours.

With the travelling salesman problem, it was exemplified how the simpler many-qudit Hamiltonian provided an advantage over the usual map of the travelling salesman problem to a quadratic unconstrained binary optimisation problem (which can use qubit-based simulators). To find the ground state of these Hamiltonians, we employed the variational Monte Carlo method with a neural quantum state, featuring solutions to the travelling salesman problem of up to 96 cities. With this, we witness one of the main contributions from machine learning to the study of quantum physics: the ability to represent a quantum state through a neural network, as we saw in chapter 3. Neural quantum states are expressive approximations of

[1] I would say that a large part of the contribution of this thesis for our group is the quantity and diversity of code (mainly written in the Python and Julia languages) with high scientific software development standards, which can seamlessly be used in the future for further research.

the quantum wave function, which serve good as variational wave functions for the variational Monte Carlo method, which allows us to handle intractable Hilbert spaces.

In this thesis, we explain how variational Monte Carlo uses neural quantum states to find ground states of Hamiltonians. However, throughout the PhD work we focused also on exploring the limitations of this approach. In the closed quantum case, we find that the method encounters difficulty very near quantum phase transitions (as all variational methods do). However, we pinpoint that trainability of neural quantum states might be the main weakness of the method. We discovered this when studying quantum dynamics as a ground state problem using the Feynman-Kitaev formalism. In this formalism, a quantum clock is added to the system of interest, and the Hamiltonian is modified in such a way that the quantum clock's state entangles with the system's state at every single discrete time of the quantum clock. We showed that this problem is particularly difficult because the probability was spread across the whole Hilbert space, making it difficult for the neural quantum state to really learn all of the wave function. Consistently, we found that neural quantum states struggled to correctly learn the ground state of this difficult problem. But further investigation revealed that neural quantum states are able to represent correctly the ground state of the Feynman-Kitaev Hamiltonian. This means that neural quantum states are expressive enough; the big problem is training them.

We also studied the Bose-Hubbard model using neural quantum states in the closed case and in the driven-dissipative case. Repeatedly we found that neural quantum states excelled at representing the ground state of the closed case, but failed in the open quantum case. To our knowledge, this is the first application of neural quantum states to open quantum systems made of bosonic particles (previous attempts were focused on spin systems only). The larger Hilbert space of a truncated bosonic system proved to be very difficult to current approaches of neural quantum states, by failing to reproduce the expected values of the number of bosons per cavity in the few-cavity regime (which is the only regime that can be compared with exact numerical calculations). Despite the current difficulties, we foresee a way forward.

The problem at hand is very difficult. If $\mathcal{L}$ is the Liouvillian of the open quantum system, one tries to find a neural quantum state ρ_{θ} that minimises $\mathrm{Tr}\left[\rho_{\theta}\mathcal{L}^{\dagger}\mathcal{L}\rho_{\theta}\right]$. Such a problem is very difficult because Liouvillian gaps can close especially near an open quantum phase transition. An alternative is time-evolution, but so far time-evolution has been restricted to neural quantum states of closed systems, and they have serious drawbacks such as not being able to correctly evolve a quantum state with a spiked probability distribution. However, it might be possible to use the Feynman-Kitaev formalism (for neural quantum states or variational circuits) to perform short time evolutions so that we can apply the theory of quantum trajectories to simulate the open quantum system. To do this, a couple of difficult long-term projects are needed. First, a Computer Algebra System needs to be developed in order to handle quantum mechanical operators as one does in the whiteboard, to allow for arbitrary symbolic manipulation of operators in quantum mechanics[2]. With this computer algebra system one is able to approximate short time evolution operators through Chebysheb operator polynomials (computing the polynomials needs the computer algebra system). Second, since the quantum trajectory theory needs the application of collapse operators onto the quantum state, we need to figure out how parameters are updated when certain collapse operators act on them. This has only been done for simple restricted Boltzmann machines that simulate quantum circuits. With these two projects, one would be able to tackle finding steady states

[2] Having a computer algebra system is not entirely necessary, but it is a key component for a flexible library that allows the study of any Hamiltonian. At the moment of writing, I am developing one such system at Zapata Computing Inc.

from a different perspective (in my opinion with good chances of success) than the one used in this thesis.

Furthermore, ground states of Hamiltonians, or in general, quantum states, naturally encode probability distributions, which are the central object in machine learning. Here we explored the other direction of the machine learning-quantum physics avenue in chapter 4. We used the mathematical framework provided by quantum mechanics to perform machine learning tasks. In particular, quantum feature maps allow the creation of quantum states from classical data, which are later processed to create quantum states that represent entire data sets. Training corresponds to preparing a quantum state. Then, we were able to perform both supervised and unsupervised machine learning tasks: classification, regression and probability density estimation. The prediction phase corresponds to further preparing a quantum state for the data point to be predicted, and a projective measurement that acts as the prediction operation. Our framework provides two main ways of doing all these steps (preparation of states and measurements): using density matrices, or using pure states as quantum states. We used density matrices for all of the applications carried out in a classical computer. Here, our framework takes the form of a quantum-inspired classical machine learning set of algorithms. We showed that our framework is competitive with many classical machine learning methods, with the impressive advantage that we can condense arbitrary probability distributions of arbitrarily large training data sets within a density matrix. On the other hand, we also used pure states in order to leverage our framework to a purely quantum machine learning framework. This means that we were able to use real quantum circuits (with the Bogotá and Lima IBM quantum computers) to perform classification and density estimation tasks on toy data sets. An important feature of this framework is that density matrices could be learnt as well, meaning that they act as modular differentiable components that can be combined with other components of the deep learning field to perform several machine learning tasks. This holds great promise in tackling challenging problems in data science such as multi-modal learning, which is the ability to learn probability distributions from data that comes from many different sources (audio, text, video, etc) in order to perform a wide variety of tasks.

In summary, we leave interesting research prospects that are naturally born out of the opportunities and difficulties encountered in the research projects carried out throughout my doctoral studies. Our contributions to the community are encapsulated in the following bullet points:

- Giant Rabi oscillation control in quantum dots embedded in acoustic cavities for the prospect of qudit-based quantum computing [Vargas-Calderón et al., 2022].

- Qudit-based approach to the travelling salesman problem solved through variational Monte Carlo and neural quantum states [Vargas-Calderón et al., 2021], with an open-source library https://gitlab.com/ml-physics-unal/htsp.

- Development of numerical tools [Vicentini et al., 2021] (https://www.netket.org/) for many-body studies to find ground states [Vargas-Calderón et al., 2020] of Hamiltonians and perform time evolution of physical systems [Vargas-Calderón et al., 2022].

- A quantum-inspired framework to perform supervised and unsupervised machine learning tasks [González et al., 2021b,a; Vargas-Calderón et al., 2022] with an open-source library for its implementation on real quantum computers https://gitlab.com/ml-physics-unal/qcm.

BIBLIOGRAPHY

Abbas, A., Sutter, D., Zoufal, C., Lucchi, A., Figalli, A., and Woerner, S. (2021). The power of quantum neural networks. *Nature Computational Science*, 1(6):403–409.

Adambukulam, C., Sewani, V., Stemp, H., Asaad, S., Madzik, M., Morello, A., and Laucht, A. (2021). An ultra-stable 1.5 t permanent magnet assembly for qubit experiments at cryogenic temperatures. *Review of Scientific Instruments*, 92(8):085106.

Akbay, M. A., Kalayci, C. B., and Polat, O. (2020). A parallel variable neighborhood search algorithm with quadratic programming for cardinality constrained portfolio optimization. *Knowledge-Based Systems*, 198:105944.

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Albert, V. V. and Jiang, L. (2014). Symmetries and conserved quantities in lindblad master equations. *Physical Review A*, 89(2):022118.

Alcazar, J., Vakili, M. G., Kalayci, C. B., and Perdomo-Ortiz, A. (2021). Geo: Enhancing combinatorial optimization with classical and quantum generative models.

Alhambra, Á. M. (2022). Quantum many-body systems in thermal equilibrium.

Anschuetz, E. R. and Kiani, B. T. (2022). Beyond barren plateaus: Quantum variational algorithms are swamped with traps.

Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., Biswas, R., Boixo, S., Brandao, F. G. S. L., Buell, D. A., Burkett, B., Chen, Y., Chen, Z., Chiaro, B., Collins, R., Courtney, W., Dunsworth, A., Farhi, E., Foxen, B., Fowler, A., Gidney, C., Giustina, M., Graff, R., Guerin, K., Habegger, S., Harrigan, M. P., Hartmann, M. J., Ho, A., Hoffmann, M., Huang, T., Humble, T. S., Isakov, S. V., Jeffrey, E., Jiang, Z., Kafri, D., Kechedzhi, K., Kelly, J., Klimov, P. V., Knysh, S., Korotkov, A., Kostritsa, F., Landhuis, D., Lindmark, M., Lucero, E., Lyakh, D., Mandrà, S., McClean, J. R., McEwen, M., Megrant, A., Mi, X., Michielsen, K., Mohseni, M., Mutus, J., Naaman, O., Neeley, M., Neill, C., Niu, M. Y., Ostby, E., Petukhov, A., Platt, J. C., Quintana, C., Rieffel, E. G., Roushan, P., Rubin, N. C., Sank, D., Satzinger, K. J., Smelyanskiy, V., Sung, K. J., Trevithick, M. D., Vainsencher, A., Villalonga, B., White, T., Yao, Z. J., Yeh, P., Zalcman, A., Neven, H., and Martinis, J. M. (2019). Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510.

Ballentine, L. E. (2014). *Quantum mechanics: a modern development*. World Scientific Publishing Company.

Barenco, A., Bennett, C. H., Cleve, R., DiVincenzo, D. P., Margolus, N., Shor, P., Sleator, T., Smolin, J. A., and Weinfurter, H. (1995). Elementary gates for quantum computation. *Phys. Rev. A*, 52:3457–3467.

Barison, S., Vicentini, F., Cirac, I., and Carleo, G. (2022). Variational dynamics as a ground-state problem on a quantum computer. *arXiv preprint arXiv:2204.03454*.

Barker, J. A. and O'Reilly, E. P. (2000). Theoretical analysis of electron-hole alignment in inas-gaas quantum dots. *Phys. Rev. B*, 61:13840–13851.

Barontini, G., Labouvie, R., Stubenrauch, F., Vogler, A., Guarrera, V., and Ott, H. (2013). Controlling the dynamics of an open many-body quantum system with localized dissipation. *Physical review letters*, 110(3):035302.

Barrett, T. D., Malyshev, A., and Lvovsky, A. I. (2022). Autoregressive neural-network wavefunctions for ab initio quantum chemistry. *Nature Machine Intelligence*, 4(4):351–358.

Batrouni, G. G. and Scalettar, R. T. (1992). World-line quantum monte carlo algorithm for a one-dimensional bose model. *Phys. Rev. B*, 46:9051–9062.

Batrouni, G. G., Scalettar, R. T., and Zimanyi, G. T. (1990). Quantum critical phenomena in one-dimensional bose systems. *Phys. Rev. Lett.*, 65:1765–1768.

Baumgartner, B. and Narnhofer, H. (2008). Analysis of quantum semigroups with gks–lindblad generators: Ii. general. *Journal of Physics A: Mathematical and Theoretical*, 41(39):395303.

Bayer, M., Ortner, G., Stern, O., Kuther, A., Gorbunov, A. A., Forchel, A., Hawrylak, P., Fafard, S., Hinzer, K., Reinecke, T. L., Walck, S. N., Reithmaier, J. P., Klopf, F., and Schäfer, F. (2002). Fine structure of neutral and charged excitons in self-assembled in(ga)as/(al)gaas quantum dots. *Phys. Rev. B*, 65:195315.

Becca, F. and Sorella, S. (2017). *Quantum Monte Carlo approaches for correlated systems*. Cambridge University Press.

Berg, E. v. d., Minev, Z. K., Kandala, A., and Temme, K. (2022). Probabilistic error cancellation with sparse pauli-lindblad models on noisy quantum processors. *arXiv preprint arXiv:2201.09866*.

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. (2011). Algorithms for hyper-parameter optimization. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.

Bharti, K., Cervera-Lierta, A., Kyaw, T. H., Haug, T., Alperin-Lea, S., Anand, A., Degroote, M., Heimonen, H., Kottmann, J. S., Menke, T., Mok, W.-K., Sim, S., Kwek, L.-C., and Aspuru-Guzik, A. (2022). Noisy intermediate-scale quantum algorithms. *Rev. Mod. Phys.*, 94:015004.

Bin, Q., Lü, X.-Y., Laussy, F. P., Nori, F., and Wu, Y. (2020). N-phonon bundle emission via the stokes process. *Physical review letters*, 124(5):053601.

Bir, G. L., Pikus, G. E., et al. (1974). *Symmetry and strain-induced effects in semiconductors*, volume 484. Wiley New York.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Blackard, J. A. and Dean, D. J. (1999). Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and Electronics in Agriculture*, 24(3):131–151.

Bloch, F. (1946). Nuclear induction. *Phys. Rev.*, 70:460–474.

Bogdanov, Y. I., Chernyavskiy, A. Y., Holevo, A., Lukichev, V. F., and Orlikovsky, A. A. (2013). Modeling of quantum noise and the quality of hardware components of quantum computers. In Orlikovsky, A. A., editor, *International Conference Micro- and Nano-Electronics 2012*, volume 8700, pages 404 – 415. International Society for Optics and Photonics, SPIE.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. (2018). JAX: composable transformations of Python+NumPy programs.

Breuer, H.-P., Petruccione, F., et al. (2002). *The theory of open quantum systems*. Oxford University Press on Demand.

Brown, R. H. and Twiss, R. Q. (1956). Correlation between photons in two coherent beams of light. *Nature*, 177(4497):27–29.

Caha, L., Landau, Z., and Nagaj, D. (2018). Clocks in feynman's computer and kitaev's local hamiltonian: Bias, gaps, idling, and pulse tuning. *Phys. Rev. A*, 97:062306.

Carleo, G., Cevolani, L., Sanchez-Palencia, L., and Holzmann, M. (2017). Unitary dynamics of strongly interacting bose gases with the time-dependent variational monte carlo method in continuous space. *Phys. Rev. X*, 7:031026.

Carleo, G., Choo, K., Hofmann, D., Smith, J. E., Westerhout, T., Alet, F., Davis, E. J., Efthymiou, S., Glasser, I., Lin, S.-H., et al. (2019a). Netket: A machine learning toolkit for many-body quantum systems. *SoftwareX*, 10:100311.

Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., Vogt-Maranto, L., and Zdeborová, L. (2019b). Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4):045002.

Carleo, G. and Troyer, M. (2017). Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606.

Carrasquilla, J. (2020). Machine learning for quantum matter. *Advances in Physics: X*, 5(1):1797528.

Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S. C., Endo, S., Fujii, K., McClean, J. R., Mitarai, K., Yuan, X., Cincio, L., and Coles, P. J. (2021). Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644.

Chang, T.-J., Meade, N., Beasley, J., and Sharaiha, Y. (2000). Heuristics for cardinality constrained portfolio optimisation. *Computers Operations Research*, 27(13):1271–1302.

Chatterjee, R. and Yu, T. (2017). Generalized coherent states, reproducing kernels, and quantum support vector machines. *Quantum Information and Communication*, 17:1292.

Chen, A., Choo, K., Astrakhantsev, N., and Neupert, T. (2022). Neural network evolution strategy for solving quantum sign structures. *Phys. Rev. Research*, 4:L022026.

Cheng, J., Wang, Z., and Pollastri, G. (2008). A neural network approach to ordinal regression. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1279–1284.

Cheng, S., Chen, J., and Wang, L. (2018). Information perspective to probabilistic modeling: Boltzmann machines versus born machines. *Entropy*, 20(8).

Choo, K., Neupert, T., and Carleo, G. (2019). Two-dimensional frustrated J_1-J_2 model studied with neural network quantum states. *Phys. Rev. B*, 100:125124.

Chow, J. M., Srinivasan, S. J., Magesan, E., Córcoles, A. D., Abraham, D. W., Gambetta, J. M., and Steffen, M. (2015). Characterizing a four-qubit planar lattice for arbitrary error detection. In Donkor, E., Pirich, A. R., and Hayduk, M., editors, *Quantum Information and Computation XIII*, volume 9500, pages 315 – 323. International Society for Optics and Photonics, SPIE.

Cirac, J. I., Pérez-García, D., Schuch, N., and Verstraete, F. (2021). Matrix product states and projected entangled pair states: Concepts, symmetries, theorems. *Rev. Mod. Phys.*, 93:045003.

Crooker, S., Barrick, T., Hollingsworth, J., and Klimov, V. (2003). Multiple temperature regimes of radiative decay in cdse nanocrystal quantum dots: Intrinsic limits to the dark-exciton lifetime. *Applied Physics Letters*, 82(17):2793–2795.

Cura, T. (2021). A rapidly converging artificial bee colony algorithm for portfolio optimization. *Knowledge-Based Systems*, 233:107505.

Daley, A. J. (2014). Quantum trajectories and open many-body quantum systems. *Advances in Physics*, 63(2):77–149.

Dalibard, J., Castin, Y., and Mølmer, K. (1992). Wave-function approach to dissipative processes in quantum optics. *Phys. Rev. Lett.*, 68:580–583.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Deng, D.-L., Li, X., and Das Sarma, S. (2017). Quantum entanglement in neural network states. *Phys. Rev. X*, 7:021021.

Deng, W.-Y., Zheng, Q.-H., Lian, S., Chen, L., and Wang, X. (2010). Ordinal extreme learning machine. *Neurocomputing*, 74(1):447–456. Artificial Brains.

Deutsch, D. and Jozsa, R. (1992). Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439(1907):553–558.

Ding, Z.-H., Cui, J.-M., Huang, Y.-F., Li, C.-F., Tu, T., and Guo, G.-C. (2019). Fast high-fidelity readout of a single trapped-ion qubit via machine-learning methods. *Phys. Rev. Applied*, 12:014038.

Dogra, N., Landini, M., Kroeger, K., Hruby, L., Donner, T., and Esslinger, T. (2019). Dissipation-induced structural instability and chiral dynamics in a quantum gas. *Science*, 366(6472):1496–1499.

Dong, D., Wu, C., Chen, C., Qi, B., Petersen, I. R., and Nori, F. (2016). Learning robust pulses for generating universal quantum gates. *Scientific Reports*, 6(1):36090.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Eisele, H., Lenz, A., Heitz, R., Timm, R., Dähne, M., Temko, Y., Suzuki, T., and Jacobi, K. (2008). Change of inas/gaas quantum dot shape and composition during capping. *Journal of Applied Physics*, 104(12):124301.

Ejima, S., Fehske, H., and Gebhard, F. (2011). Dynamic properties of the one-dimensional bose-hubbard model. *EPL (Europhysics Letters)*, 93(3):30002.

Ejima, S., Fehske, H., Gebhard, F., zu Münster, K., Knap, M., Arrigoni, E., and von der Linden, W. (2012). Characterization of mott-insulating and superfluid phases in the one-dimensional bose-hubbard model. *Phys. Rev. A*, 85:053644.

Elstner, N. and Monien, H. (1999). Dynamics and thermodynamics of the bose-hubbard model. *Phys. Rev. B*, 59:12184–12187.

Evans, D. E. (1976). Irreducible quantum dynamical semigroups. *Preprint series: Pure mathematics http://urn. nb. no/URN: NBN: no-8076*.

Feynman, R. P. (1982). Simulating physics with computers. *Int. J. Theor. Phys*, 21(6/7).

Feynman, R. P. (1985). Quantum mechanical computers. *Optics News*, 11(2):11–20.

Feynman, R. P., Vernon, F. L., and Hellwarth, R. W. (1957). Geometrical representation of the schrödinger equation for solving maser problems. *Journal of Applied Physics*, 28(1):49–52.

Fine, A. (1973). Probability and the interpretation of quantum mechanics. *The British Journal for the Philosophy of Science*, 24(1):1–37.

Fisher, M. P., Weichman, P. B., Grinstein, G., and Fisher, D. S. (1989). Boson localization and the superfluid-insulator transition. *Physical Review B*, 40(1):546–570.

Fox, A. M., Fox, M., et al. (2006). *Quantum optics: an introduction*, volume 15. Oxford university press.

Frey, B. J. (1998). *Graphical Models for Machine Learning and Digital Communication*. Adaptive Computation and Machine Learning. The MIT Press.

Frey, P. W. and Slate, D. J. (1991). Letter recognition using holland-style adaptive classifiers. *Machine Learning*, 6(2):161–182.

Gambetta, J. M., Motzoi, F., Merkel, S. T., and Wilhelm, F. K. (2011). Analytic control methods for high-fidelity unitary operations in a weakly nonlinear oscillator. *Phys. Rev. A*, 83:012308.

Gent, I. P. and Walsh, T. (1996). The tsp phase transition. *Artificial Intelligence*, 88(1-2):349–358.

Gerry, C., Knight, P., and Knight, P. L. (2005). *Introductory quantum optics*. Cambridge university press.

Giamarchi, T. (2003). *Quantum physics in one dimension*, volume 121. Clarendon press.

Gini, C. (1912). *Variabilità e mutabilità: contributo allo studio delle distribuzioni e delle relazioni statistiche.[Fasc. I.]*. Tipogr. di P. Cuppini.

Giuntini, R., Freytes, H., Park, D. K., Blank, C., Holik, F., Chow, K. L., and Sergioli, G. (2021). Quantum state discrimination for supervised classification.

Glauber, R. J. (1963). The quantum theory of optical coherence. *Phys. Rev.*, 130:2529–2539.

González, F. A., Gallego, A., Toledo-Cortés, S., and Vargas-Calderón, V. (2021a). Learning with density matrices and random features. *arXiv preprint arXiv:2102.04394*.

González, F. A., Vargas-Calderón, V., and Vinck-Posada, H. (2021b). Classification with quantum measurements. *Journal of the Physical Society of Japan*, 90(4):044002.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks.

Gorini, V., Kossakowski, A., and Sudarshan, E. C. G. (1976). Completely positive dynamical semigroups of n-level systems. *Journal of Mathematical Physics*, 17(5):821–825.

Greiner, M., Mandel, O., Rom, T., Altmeyer, A., Widera, A., Hänsch, T. W., and Bloch, I. (2003). Quantum phase transition from a superfluid to a Mott insulator in an ultracold gas of atoms. *Physica B: Condensed Matter*, 329-333:11–12.

Guvenir, H. A. and Uysal, I. (2000). Bilkent university function approximation repository.

Guyon, I., Li, J., Mader, T., Pletscher, P. A., Schneider, G., and Uhr, M. (2007). Competitive baseline methods set new standards for the nips 2003 feature selection benchmark. *Pattern Recognition Letters*, 28(12):1438–1444.

Han, Z.-Y., Wang, J., Fan, H., Wang, L., and Zhang, P. (2018). Unsupervised generative modeling using matrix product states. *Phys. Rev. X*, 8:031012.

Harrison, D. and Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1):81–102.

Hartmann, M. J. and Carleo, G. (2019). Neural-network approach to dissipative quantum many-body dynamics. *Phys. Rev. Lett.*, 122:250502.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

Hatano, N. and Petrosky, T. (2015). Eigenvalue problem of the liouvillian of open quantum systems. *AIP Conference Proceedings*, 1648(1):200005.

Hertz, J., Krogh, A., and Palmer, R. G. (1991). Introduction to the theory of neural computation. *Santa Fe Institute Studies in the Sciences of Complexity; Lecture Notes*.

Heyl, M. (2018). Dynamical quantum phase transitions: a review. *Reports on Progress in Physics*, 81(5):054001.

Heyl, M., Polkovnikov, A., and Kehrein, S. (2013). Dynamical quantum phase transitions in the transverse-field ising model. *Phys. Rev. Lett.*, 110:135704.

Hibat-Allah, M., Ganahl, M., Hayward, L. E., Melko, R. G., and Carrasquilla, J. (2020). Recurrent neural network wave functions. *Physical Review Research*, 2(2):023358.

Huang, Y. (2019). Approximating local properties by tensor network states with constant bond dimension.

Hull, J. (1994). A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554.

Jiménez-Orjuela, C., Vinck-Posada, H., and Villas-Bôas, J. M. (2018). Polarization switch in an elliptical micropillar – quantum dot system induced by a magnetic field in faraday configuration. *Physics Letters A*, 382(44):3216–3219.

Jiménez-Orjuela, C., Vinck-Posada, H., and Villas-Bôas, J. M. (2020a). Magnetic and temperature control in the emission of a quantum dot strongly coupled to a microcavity. *Physica B: Condensed Matter*, 592:412215.

Jiménez-Orjuela, C., Vinck-Posada, H., and Villas-Bôas, J. M. (2020b). Strong coupling of two quantum dots with a microcavity in the presence of an external and tilted magnetic field. *Physica B: Condensed Matter*, 585:412070.

Jiménez-Orjuela, C. A., Vinck-Posada, H., and Villas-Bôas, J. M. (2017). Dark excitons in a quantum-dot–cavity system under a tilted magnetic field. *Phys. Rev. B*, 96:125303.

Johansson, J., Nation, P., and Nori, F. (2013). Qutip 2: A python framework for the dynamics of open quantum systems. *Computer Physics Communications*, 184(4):1234–1240.

Jonsson, B., Bauer, B., and Carleo, G. (2018). Neural-network states for the classical simulation of quantum computing.

Kalayci, C. B., Ertenlice, O., Akyer, H., and Aygoren, H. (2017). An artificial bee colony algorithm with feasibility enforcement and infeasibility toleration procedures for cardinality constrained portfolio optimization. *Expert Systems with Applications*, 85:61–75.

Kalayci, C. B., Polat, O., and Akbay, M. A. (2020). An efficient hybrid metaheuristic algorithm for cardinality constrained portfolio optimization. *Swarm and Evolutionary Computation*, 54:100662.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. (2019). Analyzing and improving the image quality of stylegan.

Kashurnikov, V. A., Krasavin, A. V., and Svistunov, B. V. (1996). Mott-insulator-superfluid-liquid transition in a one-dimensional bosonic hubbard model: Quantum monte carlo method. *Journal of Experimental and Theoretical Physics Letters*, 64(2):99–104.

Kastner, M. A. (2005). Prospects for quantum dot implementation of adiabatic quantum computers for intractable problems. *Proceedings of the IEEE*, 93(10):1765–1771.

Kibler, D., Aha, D. W., and Albert, M. K. (1989). Instance-based prediction of real-valued attributes. *Computational Intelligence*, 5(2):51–57.

King, R. D., Hirst, J. D., and Sternberg, M. J. E. (1995). Comparison of artificial intelligence methods for modeling pharmaceutical qsars. *Applied Artificial Intelligence*, 9(2):213–233.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes.

Koller, W. and Dupuis, N. (2006). Variational cluster perturbation theory for bose-hubbard models. *Journal of Physics: Condensed Matter*, 18(41):9525–9540.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

Krol, A. M., Sarkar, A., Ashraf, I., Al-Ars, Z., and Bertels, K. (2022). Efficient decomposition of unitary matrices in quantum circuit compilers. *Applied Sciences*, 12(2).

Kühner, T. D. and Monien, H. (1998). Phases of the one-dimensional bose-hubbard model. *Phys. Rev. B*, 58:R14741–R14744.

Kühner, T. D., White, S. R., and Monien, H. (2000). One-dimensional bose-hubbard model with nearest-neighbor interaction. *Phys. Rev. B*, 61:12474–12489.

Labouvie, R., Santra, B., Heun, S., and Ott, H. (2016). Bistability in a driven-dissipative superfluid. *Physical review letters*, 116(23):235302.

Larry, J. and Shibin, Q. (2005). Prediction for compound activity in large drug datasets using efficient machine learning approaches.

Le Bellac, M. (2011). *Quantum physics*. Cambridge University Press.

Le Boité, A., Orso, G., and Ciuti, C. (2013). Steady-state phases and tunneling-induced instabilities in the driven dissipative bose-hubbard model. *Physical review letters*, 110(23):233601.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Li, C.-K., Roberts, R., and Yin, X. (2013). Decomposition of unitary matrices and quantum gates. *International Journal of Quantum Information*, 11(01):1350015.

Linares, M., Vinck-Posada, H., and Gómez, E. A. (2021). Magnetic control of biexcitons in a quantum dot-cavity system. *Physics Letters A*, 409:127512.

Lindblad, G. (1976). On the generators of quantum dynamical semigroups. *Communications in Mathematical Physics*, 48(2):119–130.

Lloyd, S., Schuld, M., Ijaz, A., Izaac, J., and Killoran, N. (2020). Quantum embeddings for machine learning.

Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization.

Lucas, A. (2014). Ising formulations of many np problems. *Frontiers in Physics*, 2:5.

Luo, S. L. (2005). Quantum versus classical uncertainty. *Theoretical and Mathematical Physics*, 143(2):681–688.

Lwin, K. and Qu, R. (2013). A hybrid algorithm for constrained portfolio selection problems. *Applied Intelligence*, 39(2):251–266.

Madsen, L. S., Laudenbach, F., Askarani, M. F., Rortais, F., Vincent, T., Bulmer, J. F. F., Miatto, F. M., Neuhaus, L., Helt, L. G., Collins, M. J., Lita, A. E., Gerrits, T., Nam, S. W., Vaidya, V. D., Menotti, M., Dhand, I., Vernon, Z., Quesada, N., and Lavoie, J. (2022). Quantum computational advantage with a programmable photonic processor. *Nature*, 606(7912):75–81.

Mangasarian, O. L., Street, W. N., and Wolberg, W. H. (1995). Breast cancer diagnosis and prognosis via linear programming. *Operations Research*, 43(4):570–577.

Manzano, D. and Hurtado, P. (2018). Harnessing symmetry to control quantum transport. *Advances in Physics*, 67(1):1–67.

McBrian, K., Carleo, G., and Khatami, E. (2019). Ground state phase diagram of the one-dimensional bose-hubbard model from restricted boltzmann machines. In *Journal of Physics: Conference Series*, volume 1290, page 012005. IOP Publishing.

McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R., and Neven, H. (2018). Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9(1):4812.

McClean, J. R., Parkhill, J. A., and Aspuru-Guzik, A. (2013). Feynman's clock, a new variational principle, and parallel-in-time quantum dynamics. *Proceedings of the National Academy of Sciences*, 110(41):E3901–E3909.

Minamisawa, A., Iimura, R., and Kawahara, T. (2019). High-speed sparse ising model on fpga. In *2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 670–673. IEEE.

Minganti, F., Biella, A., Bartolo, N., and Ciuti, C. (2018). Spectral theory of liouvillians for dissipative phase transitions. *Physical Review A*, 98(4):042118.

Mitarai, K., Negoro, M., Kitagawa, M., and Fujii, K. (2018). Quantum circuit learning. *Phys. Rev. A*, 98:032309.

Mollow, B. R. (1969). Power spectrum of light scattered by two-level systems. *Phys. Rev.*, 188:1969–1975.

Mølmer, K., Castin, Y., and Dalibard, J. (1993). Monte carlo wave-function method in quantum optics. *JOSA B*, 10(3):524–538.

Möttönen, M., Vartiainen, J. J., Bergholm, V., and Salomaa, M. M. (2004). Quantum circuits for general multiqubit gates. *Phys. Rev. Lett.*, 93:130502.

Motzoi, F., Gambetta, J. M., Rebentrost, P., and Wilhelm, F. K. (2009). Simple pulses for elimination of leakage in weakly nonlinear qubits. *Phys. Rev. Lett.*, 103:110501.

Mozafari, M., Jolai, F., and Tafazzoli, S. (2011). A new ipso-sa approach for cardinality constrained portfolio optimization. *International Journal of Industrial Engineering Computations*, 2(2):249–262.

Muñoz, C. S., Del Valle, E., Tudela, A. G., Müller, K., Lichtmannecker, S., Kaniber, M., Tejedor, C., Finley, J., and Laussy, F. (2014). Emitters of n-photon bundles. *Nature photonics*, 8(7):550–555.

Nagy, A. and Savona, V. (2019). Variational quantum monte carlo method with a neural-network ansatz for open quantum systems. *Physical review letters*, 122(25):250501.

Neumann, M., Kappe, F., Bracht, T. K., Cosacchi, M., Seidelmann, T., Axt, V. M., Weihs, G., and Reiter, D. E. (2021). Optical stark shift to control the dark exciton occupation of a quantum dot in a tilted magnetic field. *Phys. Rev. B*, 104:075428.

Nielsen, M. A. and Chuang, I. (2002). Quantum computation and quantum information.

Nomura, S., Segawa, Y., and Kobayashi, T. (1994). Confined excitons in a semiconductor quantum dot in a magnetic field. *Phys. Rev. B*, 49:13571–13582.

Pan, F., Chen, K., and Zhang, P. (2021). Solving the sampling problem of the sycamore quantum supremacy circuits. *arXiv preprint arXiv:2111.03011*.

Pascual Winter, M. F., Rozas, G., Fainstein, A., Jusserand, B., Perrin, B., Huynh, A., Vaccaro, P. O., and Saravanan, S. (2007). Selective optical generation of coherent acoustic nanocavity modes. *Phys. Rev. Lett.*, 98:265501.

Perea, J., Porras, D., and Tejedor, C. (2004). Dynamics of the excitations of a quantum dot in a microcavity. *Physical Review B*, 70(11):115304.

Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.-H., Zhou, X.-Q., Love, P. J., Aspuru-Guzik, A., and O'Brien, J. L. (2014). A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5(1):4213.

Petrosky, T. (2010). Complex Spectral Representation of the Liouvillian and Kinetic Theory in Nonequilibrium Physics. *Progress of Theoretical Physics*, 123(3):395–420.

Petrosky, T. and Prigogine, I. (1996). *The Liouville Space Extension of Quantum Mechanics*, pages 1–120. John Wiley & Sons, Ltd.

Pikus, G. and Bir, G. (1971). Exchange interaction in excitons in semiconductors. *Sov. Phys. JETP*, 33(1):108–114.

Pincus, M. (1970). A monte carlo method for the approximate solution of certain types of constrained optimization problems. *Operations Research*, 18(6):1225–1228.

Pino, M., Prior, J., and Clark, S. R. (2013). Capturing the re-entrant behavior of one-dimensional Bose-Hubbard model. *Physica Status Solidi (B) Basic Research*, 250(1):51–58.

Quinlan, J. R. (1993). Combining instance-based and model-based learning. In *Proceedings of the tenth international conference on machine learning*, pages 236–243.

Rahimi, A. and Recht, B. (2008a). Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184.

Rahimi, A. and Recht, B. (2008b). Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems*, volume 21, pages 1316–1323.

Roth, C. and MacDonald, A. H. (2021). Group convolutional neural networks improve quantum state accuracy. *arXiv preprint arXiv:2104.05085*.

Roy, V. (2020). Convergence diagnostics for markov chain monte carlo. *Annual Review of Statistics and Its Application*, 7(1):387–412.

Rudolph, M. S., Toussaint, N. B., Katabarwa, A., Johri, S., Peropadre, B., and Perdomo-Ortiz, A. (2022). Generation of high-resolution handwritten digits with an ion-trap quantum computer. *Phys. Rev. X*, 12:031010.

Saito, H. (2017). Solving the bose–hubbard model with machine learning. *Journal of the Physical Society of Japan*, 86(9):093001.

Schuld, M. (2021). Supervised quantum machine learning models are kernel methods.

Schuld, M. and Killoran, N. (2019). Quantum machine learning in feature hilbert spaces. *Phys. Rev. Lett.*, 122:040504.

Schuld, M. and Petruccione, F. (2018). *Supervised learning with quantum computers*, volume 17. Springer.

Schuld, M. and Petruccione, F. (2021). *Quantum Models as Kernel Methods*, pages 217–245. Springer International Publishing, Cham.

Sergioli, G., Militello, C., Rundo, L., Minafra, L., Torrisi, F., Russo, G., Chow, K. L., and Giuntini, R. (2021). A quantum-inspired classifier for clonogenic assay evaluations. *Scientific Reports*, 11(1):2830.

Sharir, O., Shashua, A., and Carleo, G. (2021). Neural tensor contractions and the expressive power of deep neural quantum states.

Shende, V. V., Bullock, S. S., and Markov, I. L. (2006). Synthesis of quantum-logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(6):1000–1010.

Shor, P. W. (1999). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 41(2):303–332.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144.

Smelyanskiy, V. N., Rieffel, E. G., Knysh, S. I., Williams, C. P., Johnson, M. W., Thom, M. C., Macready, W. G., and Pudenz, K. L. (2012). A near-term quantum computing approach for hard computational problems in space exploration. *arXiv preprint arXiv:1204.2821*.

Smolensky, P. (1986). *Information Processing in Dynamical Systems: Foundations of Harmony Theory*, page 194–281. MIT Press, Cambridge, MA, USA.

Someya, K., Ono, R., and Kawahara, T. (2016). Novel ising model using dimension-control for high-speed solver for ising machines. In *2016 14th IEEE International New Circuits and Systems Conference (NEWCAS)*, pages 1–4. IEEE.

Sorella, S., Casula, M., and Rocca, D. (2007). Weak binding between two aromatic rings: Feeling the van der waals attraction by quantum monte carlo methods. *The Journal of Chemical Physics*, 127(1):014105.

Spohn, H. (1976). Approach to equilibrium for completely positive dynamical semigroups of n-level systems. *Reports on Mathematical Physics*, 10(2):189–194.

Spohn, H. (1977). An algebraic condition for the approach to equilibrium of an open n-level system. *Letters in Mathematical Physics*, 2(1):33–38.

Steck, D. (2007). *Quantum and Atom Optics*.

Steinbach, J., Garraway, B. M., and Knight, P. L. (1995). High-order unraveling of master equations for dissipative evolution. *Phys. Rev. A*, 51:3302–3308.

Takagahara, T. (1993). Effects of dielectric confinement and electron-hole exchange interaction on excitonic states in semiconductor quantum dots. *Phys. Rev. B*, 47:4569–4584.

Takagahara, T. (2002). Theory of exciton dephasing in semiconductor quantum dots. In *Semiconductor Quantum Dots*, pages 353–388. Springer.

Tanahashi, K., Takayanagi, S., Motohashi, T., and Tanaka, S. (2019). Application of ising machines and a software development for ising machines. *Journal of the Physical Society of Japan*, 88(6):061010.

Tang, E. (2019). A quantum-inspired classical algorithm for recommendation systems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 217–228, New York, NY, USA. Association for Computing Machinery.

Tay, B. A. and Petrosky, T. (2008). Biorthonormal eigenbasis of a markovian master equation for the quantum brownian motion. *Journal of mathematical physics*, 49(11):113301.

The GPyOpt authors (2016). GPyOpt: A bayesian optimization framework in python. http://github.com/SheffieldML/GPyOpt.

Tiwari, P. and Melucci, M. (2019). Towards a quantum-inspired binary classifier. *IEEE Access*, 7:42354–42372.

Torlai, G., Mazzola, G., Carrasquilla, J., Troyer, M., Melko, R., and Carleo, G. (2018). Neural-network quantum state tomography. *Nature Physics*, 14(5):447–450.

Torlai, G. and Melko, R. G. (2018). Latent space purification via neural density operators. *Phys. Rev. Lett.*, 120:240503.

Treinish, M., Gambetta, J., Nation, P., Kassebaum, P., qiskit bot, Rodríguez, D. M., de la Puente González, S., Hu, S., Krsulich, K., Zdanski, L., Yu, J., Garrison, J., Gacon, J., McKay, D., Gomez, J., Capelluto, L., Travis-S-IBM, Marques, M., Panigrahi, A., Lishman, J., lerongil, Rahman, R. I., Wood, S., Bello, L., Singh, D., Drew, Arbel, E., Schwarm, J., Daniel, J., and George, M. (2022). Qiskit/qiskit: Qiskit 0.34.2.

Useche, D. H., Giraldo-Carvajal, A., Zuluaga-Bucheli, H. M., Jaramillo-Villegas, J. A., and González, F. A. (2021). Quantum measurement classification with qudits. *Quantum Information Processing*, 21(1):12.

Vargas-Calderón, V. (2018). Phonon-assisted tunnelling in double quantum dot molecules immersed in a microcavity. *Research Gate*.

Vargas-Calderón, V., González, F. A., and Vinck-Posada, H. (2022). Optimisation-free density estimation and classification with quantum circuits. *Quantum Machine Intelligence*, 4(2):16.

Vargas-Calderón, V., Parra-A., N., Vinck-Posada, H., and González, F. A. (2021). Many-qudit representation for the travelling salesman problem optimisation. *Journal of the Physical Society of Japan*, 90(11):114002.

Vargas-Calderón, V. and Vinck-Posada, H. (2019). Phonon-assisted tunnelling in a double quantum dot molecule immersed in a cavity. *Optik*, 183:168–173.

Vargas-Calderón, V. and Vinck-Posada, H. (2020). Light emission properties in a double quantum dot molecule immersed in a cavity: Phonon-assisted tunneling. *Physics Letters A*, 384(3):126076.

Vargas-Calderón, V., Vinck-Posada, H., and González, F. A. (2020). Phase diagram reconstruction of the bose–hubbard model with a restricted boltzmann machine wavefunction. *Journal of the Physical Society of Japan*, 89(9):094002.

Vargas-Calderón, V., Vinck-Posada, H., and Villas-Boas, J. M. (2022). Dark-exciton giant rabi oscillations with no external magnetic field. *Phys. Rev. B*, 106:035305.

Vargas-Calderón, V., Vinck-Posada, H., and González, F. A. (2022). An empirical study of quantum dynamics as a ground state problem with neural quantum states.

Vicentini, F., Biella, A., Regnault, N., and Ciuti, C. (2019). Variational neural-network ansatz for steady states in open quantum systems. *Physical review letters*, 122(25):250503.

Vicentini, F., Hofmann, D., Szabó, A., Wu, D., Roth, C., Giuliani, C., Pescia, G., Nys, J., Vargas-Calderon, V., Astrakhantsev, N., and Carleo, G. (2021). Netket 3: Machine learning toolbox for many-body quantum systems.

Vieijra, T. and Nys, J. (2021). Many-body quantum states with exact conservation of non-abelian and lattice symmetries through variational monte carlo. *Phys. Rev. B*, 104:045123.

Vivas, D. R., Madroñero, J., Bucheli, V., Gómez, L. O., and Reina, J. H. (2022). Neural-network quantum states: A systematic review.

Warren, R. H. (2013). Adapting the traveling salesman problem to an adiabatic quantum computer. *Quantum information processing*, 12(4):1781–1785.

Waugh, S. G. (1995). *Extending and benchmarking Cascade-Correlation: extensions to the Cascade-Correlation architecture and benchmarking of feed-forward supervised artificial neural networks*. PhD thesis, University of Tasmania.

Weiß, M. and Krenner, H. J. (2018). Interfacing quantum emitters with propagating surface acoustic waves. *Journal of Physics D: Applied Physics*, 51(37):373001.

Wigger, D., Weiß, M., Lienhart, M., Müller, K., Finley, J. J., Kuhn, T., Krenner, H. J., and Machnikowski, P. (2021). Resonance-fluorescence spectral dynamics of an acoustically modulated quantum dot. *Phys. Rev. Research*, 3:033197.

Wilce, A. (2021). Quantum Logic and Probability Theory. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2021 edition.

Williams, C. P. (2011). *Quantum Gates*, pages 51–122. Springer London, London.

Woods, L. M., Reinecke, T. L., and Kotlyar, R. (2004). Hole spin relaxation in quantum dots. *Phys. Rev. B*, 69:125330.

Wright, L. G. and McMahon, P. L. (2020). The capacity of quantum neural networks. In *Conference on Lasers and Electro-Optics*, page JM4G.5. Optica Publishing Group.

Wu, D., Rossi, R., and Carleo, G. (2021). Unbiased monte carlo cluster updates with autoregressive neural networks. *Phys. Rev. Research*, 3:L042024.

Wu, D., Wang, L., and Zhang, P. (2019). Solving statistical mechanics using variational autoregressive networks. *Phys. Rev. Lett.*, 122:080602.

Yang, X.-C., Yung, M.-H., and Wang, X. (2018). Neural-network-designed pulse sequences for robust control of singlet-triplet qubits. *Phys. Rev. A*, 97:042324.

Yoshioka, N. and Hamazaki, R. (2019). Constructing neural stationary states for open quantum many-body systems. *Physical Review B*, 99(21):214306.

Zahedinejad, E., Ghosh, J., and Sanders, B. C. (2016). Designing high-fidelity single-shot three-qubit gates: A machine-learning approach. *Phys. Rev. Applied*, 6:054005.

Zhao, T., De, S., Chen, B., Stokes, J., and Veerapaneni, S. (2021). Overcoming barriers to scalability in variational quantum monte carlo. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '21, New York, NY, USA. Association for Computing Machinery.

Zhong, H.-S., Wang, H., Deng, Y.-H., Chen, M.-C., Peng, L.-C., Luo, Y.-H., Qin, J., Wu, D., Ding, X., Hu, Y., Hu, P., Yang, X.-Y., Zhang, W.-J., Li, H., Li, Y., Jiang, X., Gan, L., Yang, G., You, L., Wang, Z., Li, L., Liu, N.-L., Lu, C.-Y., and Pan, J.-W. (2020). Quantum computational advantage using photons. *Science*, 370(6523):1460–1463.

Zhou, H.-Q., Orús, R., and Vidal, G. (2008). Ground state fidelity from tensor network representations. *Phys. Rev. Lett.*, 100:080601.

Zou, X. T. and Mandel, L. (1990). Photon-antibunching and sub-poissonian photon statistics. *Phys. Rev. A*, 41:475–476.

LIBERTAD Y ORDEN

INTER · AULAS · ACADEMIÆ · QUÆRE · VERUM